

DEPARTAMENTO DE TELEMÁTICA

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN
SONIDO E IMAGEN



PROYECTO FINAL DE CARRERA

DEASARROLLO DE UNA GUÍA MULTIMEDIA
SOBRE EL SISTEMA OPERATIVO ANDROID

AUTOR: ANTONIO SÁNCHEZ DE LA FUENTE RODRÍGUEZ
TUTORA: IRIA MANUELA ESTÉVEZ AYRES

Octubre de 2010

TÍTULO: *Desarrollo de una guía multimedia sobre
 el Sistema Operativo Android.*

AUTOR: *Antonio Sánchez de la Fuente Rodríguez*

TUTOR: *Iria Manuela Estévez Ayres*

La defensa del presente Proyecto Fin de Carrera se realizó el día ————— de 2010; siendo calificada por el siguiente tribunal:

PRESIDENTE: *Ricardo Romeral Ortega*

SECRETARIO *Julio Villena Román*

VOCAL *Matilde Pilar Sánchez Fernández*

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

Presidente

Secretario

Vocal

Agradecimientos

Cuando decides comenzar una travesía son muchas las rutas que te planteas seguir, muchos los lugares que te gustaría visitar, muchas las cosas que quisieras conocer; pero al final eliges un camino de entre todos los que habías estado pensando seguir durante días. Es en ese momento cuando te arriesgas a emprender un nuevo sendero.

La decisión no fue fácil, quizá por miedo a lo desconocido, quizá porque ya conocía otros caminos, o quizá fue todo fruto del azar; no lo sé, no lo recuerdo, de lo que sí estoy seguro es que si tuviera que volver a elegir una dirección lo haría por la misma senda.

Juntos hemos estado en sitios muy variados, el césped, la biblioteca, la cafetería, el callejón, la burbuja que ríe, el rey de las tortillas, Ciempozuelos, Leganés, Getafe, Colmenarejo, Madrid, Villanueva de Guadamejud, Malpartida de Plasencia, Salamanca, Ibiza o México me vienen ahora mismo a la cabeza.

Hemos compartiendo tantos y tantos momentos, y los que nos restan de estar juntos, la gran mayoría divertidos, felices, donde reinaba el buen humor; sin embargo otros eran de tensión, estrés y agobio, que casualmente se repetían de forma habitual unas tres veces al año, ahora que lo pienso es bastante curioso.

Aunque todos estos buenos ratos sólo se pueden dar cuando estás rodeado de personas que te hacen sentir bien, estás a gusto a su lado, compartes gustos y aficiones, inquietudes; y en todo momento te están proponiendo nuevas cosas que hacer en lugar de estudiar.

Estas líneas son para agradecerlos a todos los que habéis hecho posible que haya llegado hasta buen puerto. A mis padres por soportarme durante tanto tiempo; a mis hermanos por su apoyo incondicional en todo momento y ayudarme cuando lo he necesitado; a María, por devolverme la ilusión; a todos mis amigos, desde los que puedo ir andando a su casa hasta los que he necesitado un avión para ir a verlos.

Por todo lo que me habéis dado a lo largo de estos años, muchas gracias, sin vosotros no lo hubiera logrado.

Cuando hables, procura que tus palabras sean mejores que el silencio.

Proverbio Hindú

Seamos realistas y hagamos lo imposible.

Ernesto Ché Guevara

Resumen

Los teléfonos inteligentes (proveniente del anglicismo *smartphones*) han generado una revolución en la telefonía móvil, y más concretamente en las posibilidades que estos equipos son capaces de ofrecer al usuario.

Muchas de sus virtudes son debidas al Sistema Operativo (SO) que incorporan estos terminales, entre los cuales cabe destacar el SO Android, y las funcionalidades que añade a dichos teléfonos, habilitando la transmisión de audio y vídeo bajo demanda, así como intercambiar información con un servidor remoto.

Por otro lado, España se ha consolidado como uno de los destinos turísticos más importantes, logrando el tercer puesto en número de viajeros a nivel mundial, afianzando de este modo la importancia del sector en la economía del país, a la vez que se producía una diversificación de las opciones de turismo, que han dado lugar a que el turismo cultural sea el segundo tipo más solicitado por detrás del sol y la playa.

De esta forma, el objetivo del presente proyecto es generar una plataforma a través de la que cualquier persona, turista o no, que visite un espacio cultural y esté en posesión de un teléfono inteligente, pueda acceder a información remota que le sea de ayuda para la comprensión del mismo, incluyendo información sobre el propio espacio que está visitando o sobre los objetos que en él se ubican.

Durante el desarrollo del proyecto se ha diseñado la estructura de un sistema de estas características, y se ha implementado un prototipo para poder valorar la viabilidad del sistema. Dicho prototipo está compuesto por diferentes tecnologías entre las que destacar una aplicación web, servidor *HyperText Transfer Protocol* (HTTP) y aplicación sobre el SO Android.

Abstract

Smartphones have revolutionised the mobile industry by allowing end users to carry out a variety of tasks on a device beyond making calls and sending text messages.

Many of these tasks are possible due to the OS running on that the devices, among which include Google's Android OS. Features available on smart phones enable transmission of audio and video on demand, and the exchange of information with a remote server.

Spain has established itself as one of the most important tourist destinations. It has been ranked as third, in number of tourists, in the world, thereby strengthening the sector's importance to the economy. There has also been a diversification of the options available to tourists, this has resulted in cultural tourism becoming the second most requested after the sunny weather and the beach.

Thus, the objective of this project is to create a platform through which anyone, tourist or not, can visit a cultural space using a smart phone. This can be done by accessing remote information is helpful to understanding what is available in a specific cultural space as well as the nearby points of interest to the user.

The development of this project has included designing the structure of a system, and implementing a prototype to evaluate the feasibility of such a system. The prototype is composed of different technologies, most notably a web application server and HTTP server streaming audio and video.

Índice General

Resumen	VII
Abstract	IX
Índice General	XI
Lista de Figuras	XVIII
Lista de Tablas	XXII
Acrónimos	XXIII
I Primeros Pasos	1
1 Introducción al Proyecto	3
1.1 Motivaciones	5
1.2 Objetivos	6
1.3 Organización de la Memoria	7
2 Estado del Arte	9
2.1 Introducción	11
2.2 Desarrollo Web	12
2.2.1 Arquitecturas de Red	12
2.2.1.1 <i>Peer to Peer</i> P2P	13
2.2.1.2 Sistemas Distribuidos	14

2.2.1.3	Modelo Cliente Servidor	15
2.2.1.4	Clasificaciones del Modelo Cliente Servidor	16
2.2.2	Estándares web	16
2.2.2.1	Web 1.0	18
2.2.2.2	Web 2.0	19
2.2.2.3	Futuro de la Web	19
2.2.3	Lenguajes de Programación	20
2.2.3.1	Programación del lado Cliente	20
2.2.3.2	Programación del lado Servidor	22
2.2.3.3	Otros Lenguajes de Programación	24
2.3	<i>Streaming</i>	25
2.3.0.4	Protocolos	27
2.3.0.5	Tipos de <i>Streaming</i>	28
2.4	Dispositivos Móviles	29
2.4.1	Symbian	30
2.4.2	Windows Mobile	31
2.4.3	Blackberry	33
2.4.4	Palm OS	34
2.4.5	iPhone	35
2.4.6	Android	38
2.4.7	Otros SO para dispositivos reducidos	41
2.4.7.1	bada	41
2.4.7.2	MeeGo	41
2.4.7.3	LiMo	42
2.4.7.4	Java Micro Edition	42
2.4.8	Comparativa de Sistemas Operativos	42
2.4.8.1	Aspectos Básicos	43
2.4.8.2	Interacción con el Usuario	44
2.4.8.3	Desarrollo de Aplicaciones	45
2.5	Sistemas de Audioguías	46
2.5.1	Arquitectura del Sistema	46

ÍNDICE GENERAL

2.5.2	Sistemas Actuales	47
2.5.2.1	Tradicionales	48
2.5.2.2	Multimedia	48
2.5.2.3	Equipo externo	49
2.5.3	Aplicaciones en Dispositivos Móviles	49
2.5.3.1	Alhambra de Granada y The Musseum Of Modern Art	50
2.5.3.2	Toozla	50
2.5.3.3	BlueHertz	51
2.5.3.4	TourMovil	51
2.5.3.5	Buenos Aires, Argentina	51
2.5.3.6	noTours	51
2.5.3.7	Babia, YeaYe y Android Place Directory	52
2.5.3.8	wikiTude	52
2.6	Conclusiones	52
 II Diseño e Implementación		53
 3 Requisitos de la Aplicación		55
3.1	Introducción	57
3.2	Casos de Uso	57
3.2.1	Diagramas de Casos de Uso	59
3.2.2	Tablas de Casos de Uso	59
3.3	Identificación de los Requisitos	62
3.4	Requisitos de Usuario	63
3.4.1	Capacidad	63
3.4.2	Restricción	64
3.5	Requisitos de Software	65
3.5.1	Funcionales	66
3.5.2	Rendimiento	66
3.5.3	Interfaz	67
3.5.4	Recursos	68

3.5.5	Seguridad	68
3.6	Conclusiones	69
4	Implementación del Sistema	71
4.1	Introducción	73
4.2	Lenguajes de Programación	73
4.3	Elección de las herramientas	74
4.3.1	<i>Integrated Development Environment</i> (IDE) Eclipse	75
4.3.2	<i>Software Development Kit</i> (SDK) Android	76
4.3.3	XAMP	77
4.3.3.1	Apache	78
4.3.3.2	MySQL	78
4.4	Aplicación Web	79
4.4.1	Zona Pública	79
4.4.2	Zona Privada	82
4.5	Base de Datos	93
4.6	Aplicación en Dispositivo Móvil	101
4.7	Conclusiones	117
5	Verificación del Sistema	119
5.1	Introducción	121
5.2	Entorno de prueba	121
5.3	Resultados obtenidos	122
5.3.1	Parte Web	122
5.3.2	Parte Móvil	126
5.4	Conclusiones	129
III	Conclusiones del Proyecto	131
6	Conclusiones y Líneas Futuras	133
6.1	Conclusiones	135
6.2	Líneas Futuras	136

ÍNDICE GENERAL

IV	Apéndice	139
A	Presupuesto del Proyecto	141
A.1	Introducción	143
A.2	Planificación	143
A.3	Recursos Utilizados	145
A.3.1	Herramientas Software	145
A.3.2	Herramientas Hardware	145
A.3.3	Servicios	146
A.3.4	Personal	146
A.4	Presupuesto	147
B	Requisitos de Usuario	149
B.1	Introducción	151
B.2	Tablas de Requisitos de Usuario	151
B.2.1	Requisitos de Usuario de Capacidad	151
B.2.2	Requisitos de Usuario de Restricción	158
C	Requisitos de Software	165
C.1	Introducción	167
C.2	Tablas de Requisitos de Software	167
C.2.1	Requisitos de Software Funcionales	167
C.2.2	Requisitos de Software de Rendimiento	171
C.2.3	Requisitos de Software de Interfaz	174
C.2.4	Requisitos de Software de Recursos	176
C.2.5	Requisitos de Software de Seguridad	178
D	Bibliografía	181

Lista de Figuras

2.1	Esquema de funcionamiento Modelo Cliente Servidor	15
2.2	Estrategias de distribución en Sistemas Cliente Servidor	17
2.3	Lenguajes de Programación y Semántica de la Web [1]	18
2.4	Uso de Lenguajes de Programación [2]	21
2.5	Crecimiento de Lenguajes de Programación [2]	22
2.6	Ejemplo de almacenamiento mediante buffer	26
2.7	Cuota de mercado de SO móviles [3]	31
2.8	Cuota de mercado de Windows Mobile [3]	32
2.9	Precio de las Aplicaciones para Smartphones [3]	36
2.10	Volumen de Aplicaciones para Smartphones [3]	37
2.11	Arquitectura Android [4]	40
2.12	Actores de un sistema de audioguía [5]	47
2.13	Ejemplo de signoguía para personas con problemas auditivos [5]	48
3.1	Diagrama de Casos de Uso	58
4.1	Esquema del portal web 9GM	79
4.2	Diagrama de flujo del formulario de contacto	81
4.3	Diagrama de flujo para añadir o modificar usuarios	83
4.4	Diagrama de flujo para recordar contraseña	84
4.5	Diagrama de flujo para la confirmación de cuenta	85
4.6	Diagrama de flujo para menú de zona privada	86
4.7	Diagrama de flujo para comprobar la sesión	87
4.8	Diagrama de flujo para la opción contenidos	88
4.9	Diagrama de flujo para los contenidos vistos, comentados y puntuados	90
4.10	Diagrama de flujo Eliminar Usuario	91

LISTA DE FIGURAS

4.11	Diagrama de flujo para la lista de usuarios	92
4.12	Diagrama de flujo para la opción base de datos	93
4.13	Esquema de la Base de Datos de la Aplicación	94
4.14	Diagrama de flujo de la clase <i>Inicio.java</i>	102
4.15	Diagrama de flujo de la clase <i>GM.java</i>	103
4.16	Comprobación de usuario en la clase <i>GM.java</i>	104
4.17	Diagrama de flujo de la clase <i>Conexión.java</i>	105
4.18	Diagrama de flujo de la clase <i>Contenido1.java</i>	106
4.19	Diagrama de flujo de la clase <i>Contenido.java</i>	107
4.20	Diagrama de flujo de la clase <i>Información.java</i> y <i>Comentario.java</i>	108
4.21	Diagrama de flujo de la clase <i>Reproducir.java</i>	109
4.22	Diagrama de flujo de la clase <i>Puntuar.java</i>	110
4.23	Diagrama de flujo de la clase <i>Usuarios.java</i>	111
4.24	Diagrama de flujo de la clase <i>Login.java</i>	112
4.25	Diagrama de flujo de la clase <i>Datos.java</i>	113
4.26	Diagrama de flujo de la clase <i>Visto.java</i>	114
4.27	Diagrama de flujo de la clase <i>Registro.java</i>	115
4.28	Diagrama de flujo de la clase <i>Idioma.java</i>	116
4.29	Diagrama de flujo de la clase <i>Ayuda.java</i>	116
5.1	Página de inicio de la plataforma web	122
5.2	Información personal del usuario	123
5.3	Datos mostrados sobre cada contenido	124
5.4	Autologin y acceso de usuarios	125
5.5	Capturas sobre Ayuda y Salir	126
5.6	Datos Personales y Modificar Comentarios	127
5.7	Selección y Menú de Contenido	128
5.8	Información y Comentarios sobre los Contenidos	129
5.9	Reproducción y Puntuación de contenidos	130
A.1	Gráfico Gantt para la planificación del proyecto	148

Lista de Tablas

2.1	Aspectos Básicos de los SO para móviles [6]	43
2.2	Interacción con Usuario [6]	44
2.3	Desarrollo de Aplicaciones [6]	45
2.4	Visitantes del museo Thyssen de Madrid y uso de Audioguías [7]	50
3.1	Caso de Uso CU001, Ver Contenido	59
3.2	Caso de Uso CU002, Modificar Contenido	60
3.3	Caso de Uso CU003, Ver Comentarios	60
3.4	Caso de Uso CU004, Modificar Comentarios	61
3.5	Caso de Uso CU005, Puntuar Contenido	61
4.1	Tabla Usuarios	95
4.2	Tabla Comentarios	96
4.3	Tabla Estados	96
4.4	Tabla Recursos	97
4.5	Tabla Contenidos	98
4.6	Tabla Nivel	99
4.7	Tabla TipoContenido	99
4.8	Tabla Visualizaciones	100
4.9	Tabla VotacionContenido	100
A.1	Tabla de Recursos de Software	145
A.2	Tabla de Recursos de Hardware	146
A.3	Tabla de Servicios	146
A.4	Tabla de Recursos de Hardware	147
A.5	Tabla de Presupuesto Final	147

LISTA DE TABLAS

B.1	Requisitos de Usuario de Capacidad RUCW001	151
B.2	Requisitos de Usuario de Capacidad RUCW002	152
B.3	Requisitos de Usuario de Capacidad RUCW003	152
B.4	Requisitos de Usuario de Capacidad RUCW004	152
B.5	Requisitos de Usuario de Capacidad RUCW005	153
B.6	Requisitos de Usuario de Capacidad RUCW006	153
B.7	Requisitos de Usuario de Capacidad RUCW007	153
B.8	Requisitos de Usuario de Capacidad RUCW008	154
B.9	Requisitos de Usuario de Capacidad RUCW009	154
B.10	Requisitos de Usuario de Capacidad RUCM001	155
B.11	Requisitos de Usuario de Capacidad RUCM002	155
B.12	Requisitos de Usuario de Capacidad RUCM003	155
B.13	Requisitos de Usuario de Capacidad RUCM004	156
B.14	Requisitos de Usuario de Capacidad RUCM005	156
B.15	Requisitos de Usuario de Capacidad RUCM006	156
B.16	Requisitos de Usuario de Capacidad RUCM007	157
B.17	Requisitos de Usuario de Capacidad RUCM008	157
B.18	Requisitos de Usuario de Capacidad RUCM009	157
B.19	Requisitos de Usuario de Capacidad RUCM010	158
B.20	Requisitos de Usuario de Restricción RURW001	158
B.21	Requisitos de Usuario de Restricción RURW002	159
B.22	Requisitos de Usuario de Restricción RURW003	159
B.23	Requisitos de Usuario de Restricción RURW004	159
B.24	Requisitos de Usuario de Restricción RURW005	160
B.25	Requisitos de Usuario de Restricción RURW006	160
B.26	Requisitos de Usuario de Restricción RURW007	160
B.27	Requisitos de Usuario de Restricción RURM001	161
B.28	Requisitos de Usuario de Restricción RURM002	161
B.29	Requisitos de Usuario de Restricción RURM003	162
B.30	Requisitos de Usuario de Restricción RURM004	162
B.31	Requisitos de Usuario de Restricción RURM005	162

LISTA DE TABLAS

B.32	Requisitos de Usuario de Restricción RURM006	163
B.33	Requisitos de Usuario de Restricción RURM007	163
B.34	Requisitos de Usuario de Restricción RURM008	163
B.35	Requisitos de Usuario de Restricción RURM009	164
C.1	Requisitos de Software Funcionales RSFW001	167
C.2	Requisitos de Software Funcionales RSFW002	168
C.3	Requisitos de Software Funcionales RSFW003	168
C.4	Requisitos de Software Funcionales RSFW004	168
C.5	Requisitos de Software Funcionales RSFM001	169
C.6	Requisitos de Software Funcionales RSFM002	169
C.7	Requisitos de Software Funcionales RSFM003	170
C.8	Requisitos de Software Funcionales RSFM004	170
C.9	Requisitos de Software Funcionales RSFM005	170
C.10	Requisitos de Software Funcionales RSFM006	171
C.11	Requisitos de Software de Rendimiento RSRW001	171
C.12	Requisitos de Software de Rendimiento RSRW002	172
C.13	Requisitos de Software de Rendimiento RSRW003	172
C.14	Requisitos de Software de Rendimiento RSRW004	172
C.15	Requisitos de Software de Rendimiento RSWM001	173
C.16	Requisitos de Software de Rendimiento RSWM002	173
C.17	Requisitos de Software de Rendimiento RSWM003	173
C.18	Requisitos de Software de Rendimiento RSRM004	174
C.19	Requisitos de Software de Interfaz RSIW001	174
C.20	Requisitos de Software de Interfaz RSIW002	175
C.21	Requisitos de Software de Interfaz RSIM001	175
C.22	Requisitos de Software de Interfaz RSIM002	175
C.23	Requisitos de Software de Recursos RSCW001	176
C.24	Requisitos de Software de Recursos RSCW002	176
C.25	Requisitos de Software de Recursos RSCW003	177
C.26	Requisitos de Software de Recursos RSCM001	177

LISTA DE TABLAS

C.27	Requisitos de Software de Recursos RSCM002	177
C.28	Requisitos de Software de Recursos RSCM003	178
C.29	Requisitos de Software de Seguridad RSSW001	178
C.30	Requisitos de Software de Seguridad RSSW002	179
C.31	Requisitos de Software de seguridad RSSM001	179
C.32	Requisitos de Software de Seguridad RSSM002	179

AAPT <i>Android Asset Packaging Tool</i> (traducido del Herramienta de Compresión de Activos de Android)	76
ADB <i>Android Debug Bridge</i> (traducido del Puente de Depuración Android)	77
ADSL <i>Asymmetric Digital Subscriber Line</i> (traducido del Herramienta de Desarrollo Android)	121
ADT <i>Android Development Tool</i> (traducido del Línea de Abonado Digital Asimétrica)	76
AJAX <i>Asynchronous JavaScript And XML</i> (traducido del Inglés JavaScript Asíncrono y lenguaje XML)	19
API <i>Application Programming Interface</i> (traducido del Inglés Interfaz de Programación de Aplicaciones)	34
ASP <i>Active Server Pages</i> (traducido del Inglés Servidor de Páginas Activas)	23
AVM <i>Android Virtual Machine</i> (traducido del Inglés Máquina Virtual de Android)	77
CES <i>Consumer Electronic Show</i> (traducido del Inglés Feria de Electrónica de Consumo) ...	34
CGI <i>Common Gateway Interface</i> (traducido del Inglés Interfaz de Entrada Común)	22
CMS <i>Content Management System</i> (traducido del Inglés Sistemas de Gestión de Contenidos), uno de las más conocidos es Joomla	18
CSS <i>Cascade Style Sheet</i> (traducido del Inglés Hojas de Estilo en Cascada)	21
DDMS <i>Dalvik Debug Monitor System</i> (traducido del Inglés Sistema Monitor de Depuración Dalvik)	76
DNS <i>Domain Name System</i> (traducido del Inglés Sistema de Nombres de Dominio)	15
DTD <i>Document Type Definition</i> (traducido del Inglés Definición de Tipo de Documento) ..	24
ESA <i>European Space Agency</i> (traducido del Inglés Agencia Espacial Europea)	7
GPL <i>General Public License</i> (traducido del Inglés Licencia Pública General)	74
GPS <i>Global Positioning System</i> (traducido del Inglés Sistema de Posicionamiento Global) ..	5
HTML <i>HyperText Markup Language</i> (traducido del Inglés Lenguajes de Marcado de Hiper Texto)	17
HTTP <i>HyperText Transfer Protocol</i> (traducido del Inglés Protocolo de Transferencia de Hiper Texto)	VII

IDE <i>Integrated Development Environment</i> (traducido del Inglés Entorno de Desarrollo Integrado).....	XIV
IP <i>Internet Protocol</i> (traducido del Inglés Protocolo de Internet).....	12
ISO <i>International Organization for Standardization</i> (traducido del Inglés Organización Internacional para la Estandarización).....	24
J2ME <i>Java To Micro Edition</i> (traducido del Inglés Edición de Java para Dispositivos Limitados).....	42
JDT <i>Java Development Tools</i> (traducido del Inglés Herramientas para Desarrollo de Java).....	75
JSP <i>JavaServer Pages</i> (traducido del Inglés Servidor de Páginas de Java).....	23
JVM <i>Java Virtual Machine</i> (traducido del Inglés Máquina Virtual de Java).....	21
LAN <i>Local Area Network</i> (traducido del Inglés Red de Área Local).....	14
MIDP <i>Mobile Information Device Profile</i> (traducido del Inglés Perfil para Dispositivos Móviles de Información).....	34
MIME <i>Multipurpose Internet Mail Extensions</i> (traducido del Inglés Extensiones Multipropósito de Correo de Internet).....	22
NDK <i>Native Development Kit</i> (traducido del Inglés Kit de desarrollo Nativo).....	38
ONU Organización de las Naciones Unidas.....	12
OTA <i>Over The Air</i> , el usuario no interviene en la actualización del Sistema Operativo, sino que se hace de forma automática.....	44
P2P <i>Peer To Peer</i> (traducido del Inglés Red Entre Pares).....	12
PDA <i>Personal Digital Assistant</i> (traducido del Inglés Asistente Digital Personal).....	30
PDF <i>Portable Document Format</i> (traducido del Inglés Formato de Documento Portatil) ...	80
POP <i>Post Office Protocol</i> (traducido del Inglés Protocolo de Correo de Oficina).....	15
RIM <i>Research in Motion Limited</i> , también se conoce como Blackberry.....	33
RTCP <i>Real-time Transport Control Protocol</i> (traducido del Inglés Protocolo de Control de Transporte en Tiempo Real).....	28
RTP <i>Real-time Transport Protocol</i> (traducido del Inglés Protocolo de Transporte en Tiempo Real).....	27
RTSP <i>Real-time Transport Streaming Protocol</i> (traducido del Inglés Protocolo de Transporte de Streaming en Tiempo Real).....	28

ACRÓNIMOS

RSVP <i>ReSerVation Protocol</i> (traducido del Inglés Protocolo de Reserva de Recursos)	28
SDK <i>Software Development Kit</i> (traducido del Inglés Kit de Desarrollo de Software)	XIV
SGML <i>Standard Generalized Markup Language</i> (traducido del Inglés Lenguaje de Marcado Generalizado)	20
SO Sistema Operativo (del Inglés OS, <i>Operating System</i>)	VII
SDK <i>Software Development Kit</i> (traducido del Inglés Kit de Desarrollo de Software)	XIV
SQL <i>Structure Query Language</i> (traducido del Inglés Lenguaje de Consulta Estructurado) .	94
TCP <i>Transport Control Protocol</i> (traducido del Inglés Protocolo para el Control del Transporte)	27
TIC Tecnologías de la Información y la Comunicación	12
UDP <i>User Datagram Protocol</i> (traducido del Inglés Protocolo de Datagramas de Usuario) .	27
URI <i>Uniform Resource Identifier</i> (traducido del Inglés Identificador de Recurso Uniforme) .	17
URL <i>Uniform Resource Locator</i> (traducido del Inglés Localizador Uniforme de Recursos) .	80
USB <i>Universal Serial Bus</i> (traducido del Inglés Bus de Serie Universal)	5
VOD <i>Video On Demand</i> (traducido del Inglés Vídeo bajo demanda)	29
WAP <i>Wireless Application Protocol</i> (traducido del Inglés Protocolo de Aplicaciones Inalámbricas)	34
W3C <i>World Wide Web Consistorium</i> (traducido del Inglés Consistorio de la Red de Amplitud Global)	16
WWW <i>World Wide Web</i> (traducido del Inglés Red de Amplitud Mundial)	12
XAMP Distribución que instala y configura Apache, MySQL y PHP en Windows	77
XHTML <i>Extensible HyperText Markup Language</i> (traducido del Inglés Lenguaje eXtensible de Marcado de HiperTexto)	35
XML <i>Extensible Markup Language</i> (traducido del Inglés Lenguaje de Marcas eXtensible) . .	24



I

Primeros Pasos



1

Introducción al Proyecto

Una de las diferencias entre los seres humanos y los animales se basa en la comunicación, el hombre es la única parte de la naturaleza que ha desarrollado esta capacidad de forma tan extraordinaria. Tanto es así, que no nos limitamos a transmitir información a las personas que nos rodean, sino que buscamos formas de comunicación a distancias mayores, hasta el punto de que somos capaces de transferir datos al otro lado del planeta en unos pocos segundos.

CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO

1.1 Motivaciones

Las primeras computadoras que una persona podía utilizar distaban mucho de lo que hoy conocemos como ordenador personal. Aunque no podemos asegurar cual fue el primer equipo en ver la luz, seguramente su nombre pueda ser alguno de los tres siguientes: Z3 diseñado por Konrad Zuse, *Colossus* de la Agencia de Inteligencia Británica ó ENIAC creado en la Universidad de Pensilvania pueden ostentar dicho título.

Si nos fijamos en la transformación que han sufrido estas máquinas, notaremos que no sólo han adquirido una mayor capacidad de computación¹, sino que han visto reducido tanto su tamaño como su peso.

Además, los cambios que se han producido han sido tan importantes que se ha logrado dotar a unos modelos de teléfonos móviles, concretamente los teléfonos inteligentes, de una capacidad de procesamiento de datos elevada, pero sin sacrificar el concepto de movilidad que traen consigo estos terminales.

Por otro lado, los teléfonos inteligentes nos dan la posibilidad de unificar en un mismo dispositivo diferentes aplicaciones que antes estaban dispersas en varios equipos, como pueden ser el navegador *Global Positioning System* (GPS), reproductor de música, conexión a internet o una agenda de contactos.

Parte de que esto sea así es debido a los SO que estos equipos llevan instalados, los cuales les dotan de capacidad para ejecutar varios procesos al mismo tiempo, a la vez que le dan soporte para interactuar con multitud de equipamiento que los teléfonos traen consigo, como pueden ser la cámara de fotos, el módulo *bluetooth*, acelerómetros o conectores *Universal Serial Bus* (USB).

¹Ley de Moore, cofundador de *Intel*, expuso que cada 18 meses el número de transistores de un procesador se vería duplicado [8].

1.2 Objetivos

El objetivo del presente proyecto es generar un sistema de Guía Multimedia para aquellos dispositivos móviles que integren el SO Android, pudiendo soportar como contenidos la mayor parte posible de elementos de interés turístico, utilizando las herramientas que supongan el menor coste para el sistema final.

Una de las premisas del desarrollo persigue dotar a la aplicación de la mayor ubicuidad posible, pudiendo conectarse remotamente a la misma tanto para labores de administración como de uso.

Por ello, se creará una aplicación web mediante la cual administrar los contenidos que ofrece la aplicación, además de un portal web del producto final para su navegación a través de internet.

Por otro lado, se pretende desarrollar una aplicación que permita a los agentes turísticos independizarse del uso y alquiler de hardware a los usuarios, pasando a utilizar éstos su teléfono móvil inteligente para dicha tarea.

El desarrollo llevará a cabo el soporte necesario para realizar la reproducción de contenidos remotos con un tiempo de espera reducido, dando la posibilidad de ver textos, imágenes y vídeos, así como reproducir audios.

En resumen, el citado sistema se dividirá en dos partes bien diferenciadas, que por otro lado serán necesarias para el correcto funcionamiento del mismo:

- Parte Web

- Portal web para usuarios

- Aplicación Web de Gestión

- Parte Móvil

- Aplicación Android

CAPÍTULO 1. INTRODUCCIÓN AL PROYECTO

1.3 Organización de la Memoria

La memoria se ha dividido en cuatro partes claramente diferenciadas, en primer lugar encontramos los *Primeros Pasos*, donde están los capítulos de *Introducción al Proyecto* y el *Estado del Arte*.

El capítulo de *Introducción al Proyecto* nos acerca las motivaciones que han dado lugar a la realización del presente proyecto, así como los objetivos que se han marcado al comienzo del mismo, finalizando con una primera aproximación a las distintas partes y capítulos que componen esta memoria.

Por su parte, el *Estado del Arte* nos acerca la actualidad del contexto en el que se enmarca el presente proyecto, que no es otro que el desarrollo web, las tecnologías de *Streaming*, los dispositivos móviles y los sistemas de audioguías.

A continuación nos encontramos la parte de *Diseño e Implementación*, que alberga los capítulos de *Requisitos de la Aplicación*, *Implementación del Sistema* y *Verificación del Sistema*.

Debido a que se sigue el estándar de la *European Space Agency* (ESA), se hace necesario el estudio y redacción de los requisitos de usuario, requisitos de software y los diferentes casos de uso que tendrá la aplicación, desarrollándose todos ellos en el capítulo *Requisitos de la Aplicación*.

En el capítulo *Implementación del Sistema* nos adentraremos en el desarrollo del sistema, comenzando por las diferentes herramientas utilizadas para la realización del mismo, explicando a continuación cómo se ha logrado implementar las funcionalidades que el programa requiere, tanto en la parte web como en la parte *Android*, para finalizar explicando la base de datos que se ha desarrollado.

Una vez realizado el desarrollo del sistema, comprobaremos que el mismo funciona adecuadamente, y realiza las labores que se han descrito en los diferentes requisitos, repasándose cada función de la aplicación en el capítulo *Verificación del Sistema*.

En la siguiente parte, *Conclusiones del Proyecto*, está incluido el capítulo de *Conclusiones y Líneas Futuras*, donde se reflejan las diferentes conclusiones a las que se han llegado una vez

1.3. ORGANIZACIÓN DE LA MEMORIA

finalizado el proyecto, así como las posibles líneas futuras de trabajo que pueden realizarse a la aplicación.

Por último encontramos la parte de *Apéndices*, donde tenemos el *Presupuesto del Proyecto*, se desarrollan los diferentes *Requisitos de Usuario* y *Requisitos de Software*, y la *Bibliografía* que se ha consultado para la consecución del proyecto.



2

Estado del Arte

Como animal gregario, el ser humano tiene la necesidad de comunicarse, bien sea con una persona que esté próxima a él o bien con alguien alejado, de manera que se ha consolidado entre los pilares fundamentales de su existencia el hecho de transmitir información a los demás, siendo la comunicación algo esencial para el ser humano.

Multitud de ejemplos de comunicación a larga distancia encontramos a lo largo de los años, tambores o señales de humo pueden llevarnos a civilizaciones lejanas, pero también podemos encontrar formas de transmisión de información más próxima a nuestro tiempo, como es el lenguaje de los silbidos que aún se practica en La Gomera o en Los Chinantecos (México).

CAPÍTULO 2. ESTADO DEL ARTE

2.1 Introducción

En el presente capítulo se hará una breve descripción del estado actual de las diferentes tecnologías que confluyen en este Proyecto y son necesarias para su realización:

- Desarrollo Web

Comenzaremos introduciendo diferentes aspectos del desarrollo web, que usaremos para llevar a cabo esta aplicación.

Así pues, veremos distintas arquitecturas de red, profundizando en el modelo cliente servidor, para después describir los diferentes estándares web.

Para finalizar este apartado, hablaremos de los lenguajes de programación que nos atañen, dividiéndolos entre los lenguajes del lado del servidor y los lenguajes del lado del cliente, además de otros, que por su importancia también trataremos.

- *Streaming*

A continuación nos centraremos en la tecnología *streaming*, con lo que enunciaremos los tipos que se dan en la actualidad, así como los protocolos que se utilizan para llevarlos a cabo.

- Dispositivos Móviles

Por otro lado, estudiaremos los SO para Dispositivos Móviles, y en concreto los que se instalan en los teléfonos *smartphones*, y por ello realizaremos un breve repaso a las características fundamentales de estos SO, comparando las diferentes funcionalidades que cada uno de ellos presenta, así como las características que los hacen sobresalir respecto del resto.

- Sistemas de Audioguías

No hay que olvidar, que nuestro objetivo final es elaborar un sistema de Guía Multimedia, similar a los que se presentan en museos o galerías de arte, y es por ello que estudiaremos cual es la arquitectura típica de estos equipos, así como las opciones que encontramos en el mercado actualmente, prestando especial atención a las aplicaciones que ya han sido desarrolladas para teléfonos móviles.

2.2 Desarrollo Web

Hablar de desarrollo web es hacerlo de Internet, término que a veces es confundido con *World Wide Web* (WWW), mientras Internet hace referencia a un conjunto de máquinas, principalmente ordenadores, que a través del protocolo de comunicaciones *Internet Protocol* (IP) se interconectan entre sí, WWW es un servicio que ofrece internet, siendo éste el conjunto de documentos de hipertexto que utiliza internet para llegar a los usuarios finales [9].

Junto a estas definiciones anteriores, añadir que el desarrollo web dará como resultado un soporte para la transmisión de información, por medio de equipos de almacenamiento, administración, transmisión y búsqueda, conociéndose a estas tecnologías como Tecnologías de la Información y la Comunicación (TIC) [10].

Estas tecnologías están adquiriendo una importancia en nuestras vidas cada vez mayor, tal y como muestran las palabras de Kofi Annan (Exsecretario General de la Organización de las Naciones Unidas (ONU)) en el 'Informe del Milenio' en el año 2002 [10]:

"Las nuevas tecnologías brindan una oportunidad sin precedentes de que los países en desarrollo se salten las primeras fases del desarrollo. Hay que hacer todo lo posible para maximizar el acceso de sus pueblos a las nuevas redes de información." [10]

Además de todo lo anterior, para conocer con mayor profundidad el desarrollo web, hablaremos de los tipos de arquitecturas de red que nos podemos encontrar, continuando con los diferentes estándares web, para finalizar describiendo lenguajes de programación que hacen posible el desarrollo web.

■ ■ 2.2.1 Arquitecturas de Red

A continuación repasaremos diversas arquitecturas de red, como son *Peer To Peer* (P2P), sistemas distribuidos y el modelo cliente servidor.

CAPÍTULO 2. ESTADO DEL ARTE

■ ■ ■ 2.2.1.1 *Peer to Peer* P2P

Este tipo de redes se caracteriza por no presentar ordenadores que actúen como clientes ni como servidor, sino que lo hacen de forma simultánea respecto al resto de nodos de la red, un ejemplo de este tipo de redes es *Napster* [11].

Estas redes son escalables, cuantos más nodos tenga una red P2P su funcionamiento teóricamente será mejor; robustas, al haber varios nodos con la información demandada, esta red se puede reponer de una caída de parte de la red; descentralizadas, al ser todos los nodos iguales, no encontramos ninguno imprescindible; y con costes distribuidos entre todos los nodos de la red [11].

Aunque también presentan algunos problemas, entre los que destacan la baja implementación en seguridad de estas redes, así como la identificación de nodos que no posean una dirección IP pública, además del debate abierto en torno a la legalidad de compartir información sujeta a derechos de autor mediante estas redes [11].

Por último haremos una clasificación [11]:

- Centralizadas

Existe un único servidor que realiza labores de enlace para todos los nodos y mantiene de forma dinámica la lista de dónde se encuentra la información, no obstante, se ve comprometida la escalabilidad de la red y el anonimato de los nodos, al tener un único servidor y almacenar información sobre dónde está hospedada la información.

En algunas ocasiones, pueden existir más de un servidor, donde estos almacenan únicamente información sobre los recursos de la red, sin hospedar nada de información, labor que realizarán los nodos, se les conoce como redes híbridas o mixtas.

- Distribuidas

Aquí todos los usuarios realizan labores de cliente y servidor, realizándose las conexiones siempre entre usuarios, la información sobre los nodos, así como la información almacenada es mantenida por todos los usuarios de la red, aumentando la escalabilidad y la robustez del sistema.

2.2.1.2 Sistemas Distribuidos

Un sistema distribuido se define como una colección de ordenadores conectados por una red de comunicaciones, donde el usuario percibe como un único sistema. El acceso del usuario a los recursos remotos es de la misma manera en que lo haría a recursos locales, además estos sistemas deben soportar que un componente del sistema se caiga, reemplazándolo otro componente [12].

El tamaño de un sistema distribuido puede ser muy variado, ya sean decenas de equipos en una red *Local Area Network* (LAN) hasta miles o millones de equipos que forman Internet [12].

Las características principales que presentan estos sistemas son [12]:

- **Compartición de Recursos**

Un sistema de este tipo comparte muy diversas entidades, desde discos duros o impresoras hasta ficheros o bases de datos, por ello se hace necesaria la presencia de un software específico que a través de la interfaz de red gestione la compartición de estos recursos, que tendrá en cuenta políticas, métodos específicos y requisitos de los usuarios de la red.

- **Apertura**

Un sistema puede ser abierto o cerrado en función de la capacidad del mismo para añadir nuevas extensiones *hardware* y *software*. En los sistemas distribuidos abiertos los componentes pueden provenir de fabricantes heterogéneos siempre que cumplan los estándares requeridos.

- **Concurrencia**

En un sistema distribuido la posibilidad de ejecución en paralelo de procesos ocurre debido a que muchos usuarios interactúan simultáneamente.

- **Escalabilidad**

Estos sistemas trabajan de manera eficiente a diferentes escalas, por ello, tanto el software del sistema como el de aplicación no deberían sufrir cambios al variar la escala del sistema. La demanda de escalabilidad en los sistemas distribuidos ha conducido a una filosofía de diseño en que cualquier recurso simple *-hardware o software-* puede extenderse para proporcionar servicio a tantos usuarios como se quiera, aunque si la demanda de un recurso

CAPÍTULO 2. ESTADO DEL ARTE

crece se debería extender el sistema para ofrecer un mejor servicio.

- Tolerancia a errores

Para evitar que un fallo paralice el sistema se implementan dos puntos complementarios: Redundancia *hardware* (uso de componentes redundantes) y recuperación del software (diseño de programas que sean capaces de recuperarse de los fallos).

- Transparencia

La transparencia en los sistemas distribuidos se da debido a la concepción que el usuario y desarrollador tienen del sistema como un único elemento. Se definen varias formas de transparencia en estos sistemas: De acceso, de localización, de concurrencia, de replicación, de fallos, de migración, de prestaciones y de escalado.

■ ■ ■ 2.2.1.3 Modelo Cliente Servidor

Este modelo basa su funcionamiento en un cliente que realiza peticiones, y un servidor que a partir de las peticiones que le llegan ofrece respuestas, estando separados ambos de manera lógica, pudiendo estar en el mismo equipo o en máquinas diferentes [12]. Podemos ver un ejemplo de esta



Figura 2.1: Esquema de funcionamiento Modelo Cliente Servidor

arquitectura en la Figura 2.1, donde el cliente hace el papel de parte activa, realizando peticiones al servidor, y esperando recibir la respuesta. Habitualmente nos conectamos a varios servidores a la vez, como puede ser *Domain Name System* (DNS), HTTP o *Post Office Protocol* (POP) [12].

Por su parte, el servidor actúa de forma pasiva, esperando las peticiones que realizan los

clientes, procesándolas y devolviendo la respuesta de cada una de ellas. Además acepta multitud de conexiones desde sitios muy dispares [12].

Entre las ventajas que ofrece este modelo podemos encontrar la centralización del control, pudiendo evitar fallos por un cliente malicioso; integración de sistemas diferentes más potentes con otros más humildes; o agilizar la actualización de información. Además, es un modelo escalable, ya que podemos aumentar capacidades de cliente y de servidor por separado, o aumentar el número de clientes o servidores [12].

El mantenimiento se hace más sencillo, debido a que la modificación de localización de un servidor es transparente a los usuarios, así como las actualizaciones o reparaciones del servidor que no afectarán a los clientes [12].

Por otro lado, existen problemas de centralización de peticiones y robustez en el servidor, algo que se puede solventar duplicando el servidor y la información almacenada en el mismo [12].

■ ■ ■ 2.2.1.4 Clasificaciones del Modelo Cliente Servidor

Podemos clasificar estos sistemas en base a diferentes criterios, por ejemplo en función del servicio que ofrecen, encontrando aquí servidores de archivos, bases de datos, transacciones, objetos web [12].

También es usual verlo desde el punto de vista de la distribución de aplicaciones, tal y como muestra la Figura 2.2, distinguiendo la opción de cliente pesado/servidor ligero, típica de los servidores de archivos y bases de datos, donde la aplicación corre en el lado cliente principalmente y el servidor simplemente devuelve información en bruto; y cliente ligero/servidor pesado, en implementaciones de servidor web y de transacciones, donde el cliente actúa de interfaz de usuario, y el servidor se encarga de la mayor parte de la aplicación [12].

■ ■ 2.2.2 Estándares web

Los Estándares Web son un conjunto de especificaciones y recomendaciones técnicas generadas por el *World Wide Web Consistorium* (W3C) (organismo neutro que desarrolla los

CAPÍTULO 2. ESTADO DEL ARTE

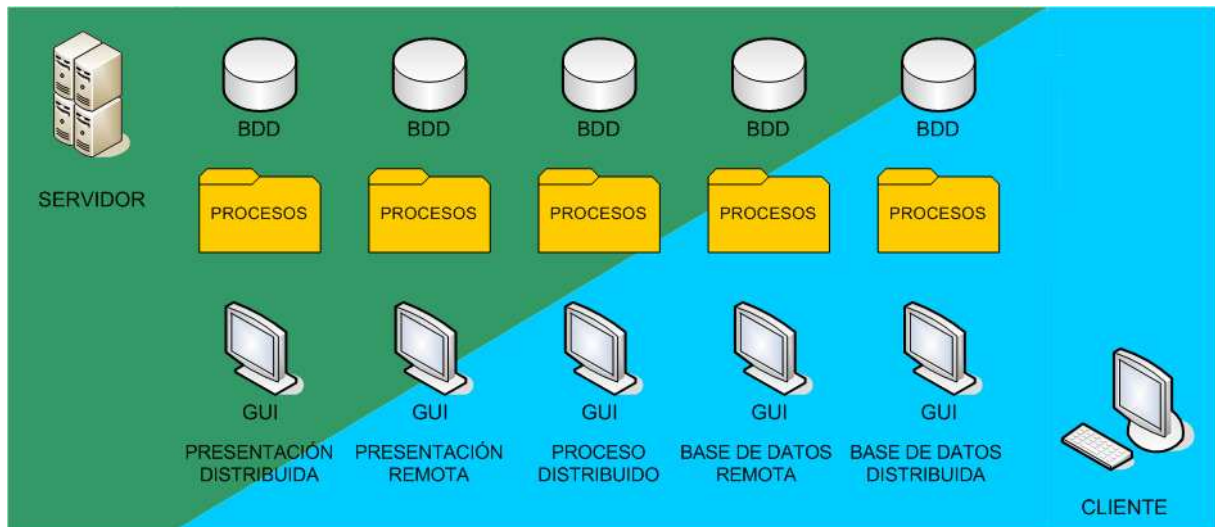


Figura 2.2: Estrategias de distribución en Sistemas Cliente Servidor

estándares para la web para poder guiar a esta hacia su máximo potencial) [13] que definen mejores prácticas para la construcción de sitios de Internet [14].

Entre los estándares web que existen, ver Figura 2.3, tres de ellos son esenciales para el funcionamiento de WWW: *Uniform Resource Identifier* (URI), HTTP e *HyperText Markup Language* (HTML). Para realizar transacciones en Internet podemos utilizar múltiples protocolos, entre todos ellos destacamos HTTP, el cuál define la sintaxis y la semántica de los elementos de esta arquitectura [14].

Para poder acceder a un recurso es necesario que éste sea identificado de forma unívoca. El protocolo URI realiza esta labor a través de una cadena corta de caracteres que puede estar formada por el protocolo de acceso al recurso, el dominio, la ruta y consulta en caso de ser necesaria [14].

Una vez se ha accedido a los datos y se ha transportado la información hasta el equipo cliente, necesitamos presentarlos al usuario final, para ello se utiliza un lenguaje para que el navegador interprete y presente la información, ese lenguaje de programación es HTML [14].

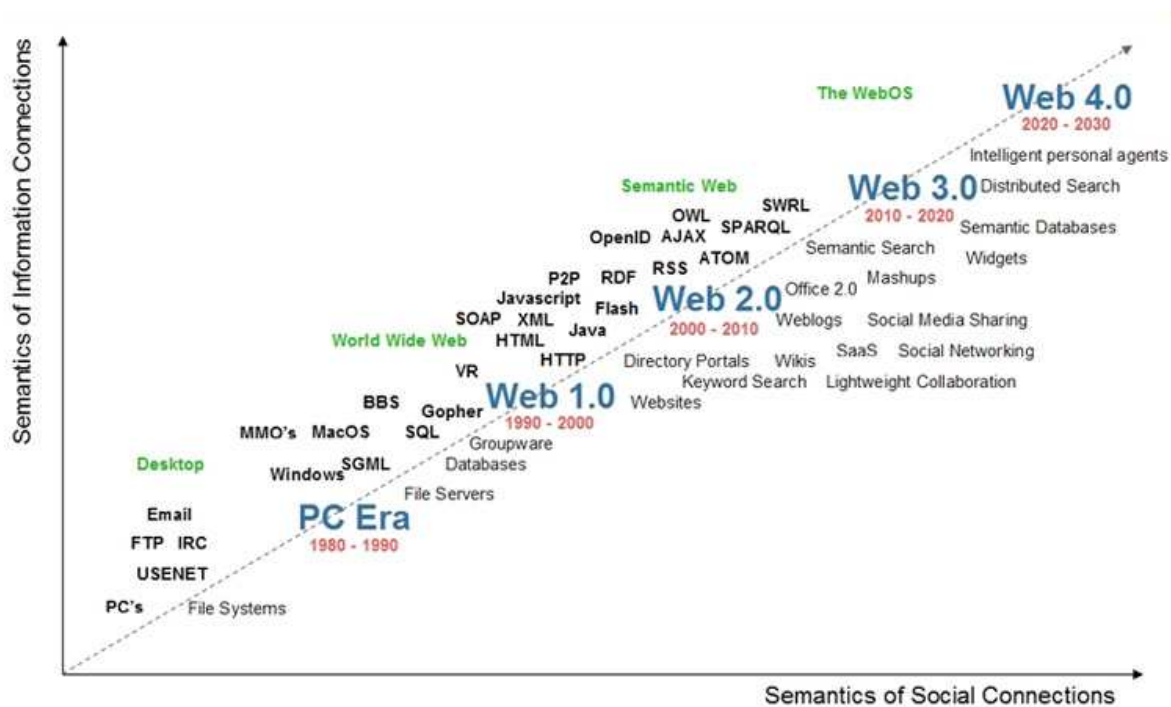


Figura 2.3: Lenguajes de Programación y Semántica de la Web [1]

2.2.2.1 Web 1.0

La palabra que mejor describe esta parte del WWW es "estática", aunque la mayor revolución que se produjo fue en materia de publicidad, cayendo en el *webmaster* la responsabilidad de diseñar, generar los contenidos y hacer accesible la web, con lo que sus conocimientos debían ser heterogéneos. Las opciones del lado del servidor eran poco potentes, y los usuarios únicamente se comunicaban con el administrador por medio de enlaces, correo electrónico o formularios web [14].

Un poco más adelante se popularizaron los *Content Management System* (CMS), y con ello lo que hoy conocemos como la web dinámica o web 1.5, o lo que es lo mismo la capacidad de brindar una nueva forma de publicar contenidos en WWW. A partir de una base de datos que mantenemos actualizada y un mayor procesamiento del lado del servidor conseguimos lanzar al cliente páginas actualizadas, haciendo uso de tecnologías como PHP o SQL, pero el problema seguía siendo que el *webmaster* era el único capaz de modificar los contenidos [14].

CAPÍTULO 2. ESTADO DEL ARTE

■ ■ ■ 2.2.2.2 Web 2.0

"*La Web 2.0 es la transición de las páginas tradicionales a las aplicaciones Web orientadas a los usuarios*", esas fueron las palabras que Dale Dougherty pronunció en octubre de 2004 y que dieron paso al concepto Web 2.0 [14].

La web 2.0 se basa en la colaboración como medio de crecimiento de sus contenidos, donde la figura del administrador únicamente realiza labores de supervisión y reparación de problemas derivados del funcionamiento, como caídas del servidor o errores en la programación. Podemos encontrar ejemplos en webs como *Wikipedia*, *Delicious* o *Flickr*. Además destacar la incorporación de *Asynchronous JavaScript And XML* (AJAX) a la programación [14].

Por tanto, con la Web 2.0 hemos pasado de usar taxonomías (clasificar por medio de directorios) a folksonomías (uso de etiquetas para la clasificación, que pueden llevar a errores si no se controla al dar la posibilidad de aparecer sinónimos, homonimia o polisemia), webs personales a *blogs*, publicar contenidos a participar en los mismos, contenidos modificados por el webmaster a dejar a la comunidad que vaya añadiendo, modificando y eliminando los datos [14].

■ ■ ■ 2.2.2.3 Futuro de la Web

Podemos definir la Web 3.0 como un espacio que trata de conocer a cada cliente para poder adaptarse a sus particularidades. Constituye la próxima evolución de Internet, recibiendo además el nombre de Web semántica, pues facilita y simplifica las búsquedas haciendo uso de una infraestructura común con la que compartir, procesar y transferir información [14].

El avance se está produciendo gracias a la folksonomía; esto es, la transición de la Web 2.0 a la 3.0, basada en los *tag* o etiquetas. Así, las redes sociales y los principales medios de comunicación se están introduciendo en este proceso, haciendo de sus páginas espacios interactivos donde el internauta utiliza diferentes aplicaciones [14].

Dale Dougherty, creador del término Web 3.0, ofrece algunos ejemplos de aplicaciones de esta nueva plataforma informática como *Google Earth* o las diferentes estructuras de funcionamiento del *iPhone*. Para Dougherty éste es en definitiva un lugar en la Web destinado a estructurar y

mostrar cómo puede funcionar la comunidad de internautas de forma conjunta y eficaz [14].

Tim Berners-Lee, creador de World Wide Web, considera que la Web semántica revolucionará la red eliminando las limitaciones que hoy en día padecen los usuarios. Para él, la Web 3.0 es la Web 2.0 sin barreras, y su cometido debe ser el de convertir la información en conocimiento mediante la ordenación y clasificación de los contenidos, en modo tal que el ordenador los interprete y tome decisiones gracias al cruce de datos [14].

■ ■ 2.2.3 Lenguajes de Programación

Existen multitud de lenguajes de programación, ver Figura 2.4 y Figura 2.5, utilizados en diversos aspectos del desarrollo, por ello realizaremos un repaso a diferentes lenguajes de programación utilizados en el desarrollo web. Comenzaremos hablando de los lenguajes usados en el lado del cliente, para continuar con los utilizados en el lado del servidor, además comentaremos otros lenguajes de programación que son altamente utilizados en el desarrollo web.

■ ■ ■ 2.2.3.1 Programación del lado Cliente

- HTML

HTML es una aplicación directa de *Standard Generalized Markup Language* (SGML) (estándar internacional para la especificación de lenguajes de marcas de carácter general), el cual nos permite, de manera sencilla, describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otras fuentes de información relacionadas, y con inserciones multimedia. La descripción se basa en especificar en el texto la estructura lógica del contenido así como los diferentes efectos que se quieren dar y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado como puede ser un navegador web [15].

- JavaScript

Definido por algunos expertos como el lenguaje de script de la web, interpretado, multiplataforma y orientado a objetos. Creado por *Netscape* para utilizarlo en el desarrollo de

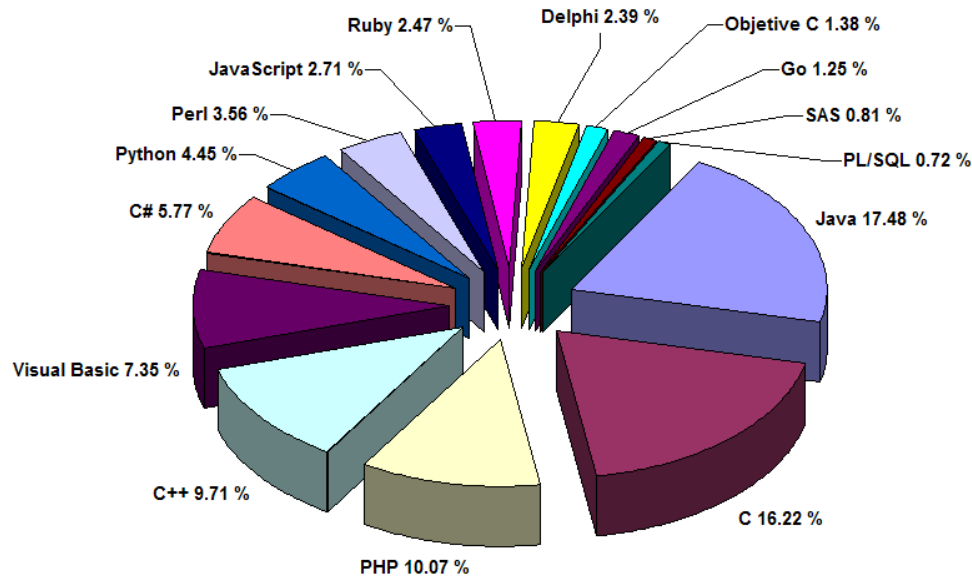


Figura 2.4: Uso de Lenguajes de Programación [2]

sitios web. Proporciona un control dinámico sobre las páginas en las que está escrito [16].

- Applets de Java

Como su nombre indica, consiste en un *script* programado en Java que podemos visualizar a través la *Java Virtual Machine* (JVM), el código permanece embebido dentro del texto HTML y nos ofrece diferentes ventajas como ser multiplataforma, poseer una seguridad alta, trasladar trabajo del servidor al cliente o ser soportado por la mayoría de los navegadores [17].

- Flash

Entorno de desarrollo de contenido interactivo soportado en diferentes plataformas y navegadores, actualmente es distribuido por Adobe, y cuenta con características como gráficos vectoriales, interacción con el usuario, sonido o vídeo [18].

- *Cascade Style Sheet* (CSS)

Lenguaje que permite al desarrollador tener el control sobre cómo se va a mostrar un contenido en concreto, logrando separar de esta forma contenido de presentación. Aunque en nuestro caso lo usaremos junto al código HTML, se puede ampliar a otros aspectos como el de impresión o de *Text To Speech* [19].

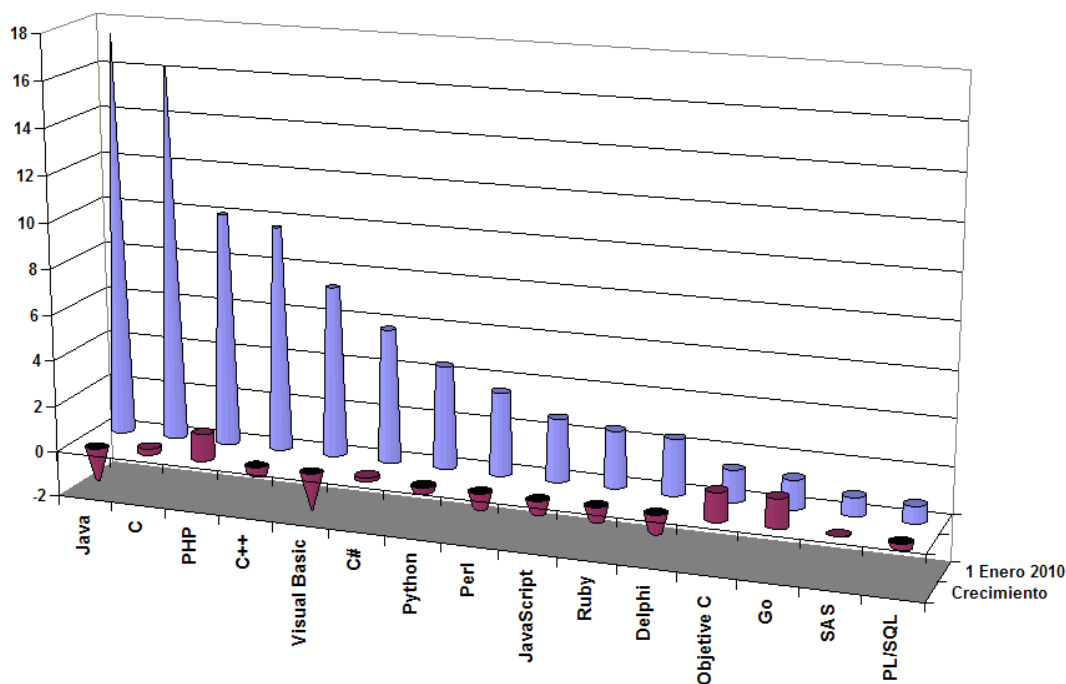


Figura 2.5: Crecimiento de Lenguajes de Programación [2]

2.2.3.2 Programación del lado Servidor

- *Common Gateway Interface* (CGI)

Este lenguaje de programación fue utilizado para programar las primeras web dinámicas, define una manera de comunicar al servidor con un programa externo que da respuesta a la petición del cliente, devolviendo un objeto *Multipurpose Internet Mail Extensions* (MIME), que será devuelto al usuario [20].

CAPÍTULO 2. ESTADO DEL ARTE

- Perl

Implementado por Larry Wall en 1987, *Perl* es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como *Javascript* o *Active Server Pages* (ASP). Esto quiere decir que el código de los *scripts* en *Perl* no se compila sino que cada vez que se quiere ejecutar se lee el código y se pone en marcha interpretando lo que hay escrito. Además es extensible a partir de otros lenguajes, ya que desde *Perl* podremos hacer llamadas a subprogramas escritos en otros lenguajes. También desde otros lenguajes podremos ejecutar código *Perl* [21].

- Python

Creado por Guido van Rossum, su nombre proviene de la aficción del autor por el grupo Monty Python. Es un lenguaje de programación interpretado de código abierto, permite dividir la programación en partes reutilizables por otros programas, incluye una biblioteca de módulos estándar, entre los que se incluyen aquellos que dan funcionalidad de llamadas al sistema o sockets. Existen módulos creados para la programación de dispositivos móviles, como por ejemplo PyS60, *Python* para *Symbian S60* [22].

- ASP

Desarrollado por *Microsoft* para programar páginas dinámicas, sigue la línea de programación de otros lenguajes del fabricante de software como *C#* o *Visual Basic* [23].

- PHP

Solución libre y gratuita para ASP, lenguaje de programación interpretado, y a partir de la versión 5.0 también orientado a objetos, creado por Rasmus Lerdorf en 1994. Su uso es muy elevado, situándose tras Java y C, además posee una gran similitud con el lenguaje C, lo que lleva a los programadores familiarizados con este lenguaje de programación una curva de aprendizaje rápida. Otra característica importante de PHP es que permite la conexión a bases de datos del tipo SQL [24].

- *JavaServer Pages* (JSP)

Tecnología basada en el lenguaje Java que permite incorporar contenido dinámico a las páginas web. Los archivos JSP combinan HTML con etiquetas especiales y fragmentos de código Java [25].

Las partes más importantes del código de una página JSP son [25]:

1. Directivas

Indican información general de la página, como puede ser importación de clases, página a invocar ante errores, si la página forma parte de una sesión, etc.

2. Declaraciones

Sirven para declarar métodos o variables.

3. Scriptlets

Código Java embebido.

4. Expresiones

Expresiones Java que se evalúan y se envían a la salida.

5. Tags JSP

Etiquetas especiales que interpreta el servidor.

Quizá la ventaja fundamental de JSP es que tenemos toda la potencia del lenguaje Java a nuestro alcance, con sus ventajas como reusabilidad, robustez o capacidad de ejecución multiplataforma [25].

- Ruby

Distribuido actualmente bajo una licencia de Software Libre, fue publicado por Yukihiro Matsumoto en 1995. Lenguaje orientado a objetos, combina sintaxis de otros lenguajes como *Perl* o *Python* y funcionalidades de *Lisp* o *Dylan*. Existe un entorno de programación llamado *Ruby On Rails* que ayuda a la elaboración de aplicaciones web con acceso a base de datos [26].

2.2.3.3 Otros Lenguajes de Programación

- *Extensible Markup Language* (XML)

Al igual que HTML, este lenguaje proviene de un lenguaje creado por IBM, SGML, y más tarde estandarizado por la *International Organization for Standardization* (ISO), este lenguaje busca dar una solución al problema de expresar información de forma estructurada, complementándose con otros lenguajes como *Document Type Definition* (DTD) [27].

CAPÍTULO 2. ESTADO DEL ARTE

- SQL

Lenguaje de acceso a base de datos, lenguaje declarativo con el que mediante su programación indicamos que queremos hacer, y no como hacerlo, ejemplos de bases de datos que hacen uso de este lenguaje son *mySQL* o *PostgreSQL* [28].

2.3 Streaming

Para poder reproducir un contenido multimedia que no se encuentra en nuestro ordenador, podemos escoger diferentes formas de hacerlo, por un lado podemos descargar el archivo, para una vez completada la descarga reproducirlo; por otro lado podemos realizar una descarga progresiva, donde el archivo se almacenará en nuestro equipo mientras se va reproduciendo una vez cargado el *buffer* necesario; o podemos realizarlo mediante *Streaming*, es decir, reproduciremos el contenido sin necesidad de almacenar en nuestro equipo todo el fichero, haciendo también uso de un *buffer* para la correcta reproducción [29].

Por otro lado, el *buffer* juega un papel importante en este tipo de comunicación, y para explicarlo usaremos un símil con un cubo, como muestra la Figura 2.6, cuando comenzamos la reproducción los paquetes que nos llegan comienzan a llenar este cubo, cuando llega a un nivel determinado comienza la reproducción, mientras continua llenándose el cubo por el flujo de datos, este sistema nos permite superar de manera momentánea pérdidas de paquetes, colapsos en la red, o cualquier otra situación que lleve a que el flujo de datos se vea alterado, con lo que usaremos el contenido que ya hemos almacenado [29].

Además, juegan un papel importante los *codecs* de audio y vídeo que permiten realizar una compresión de los datos a enviar, permitiendo transportar en el mismo tamaño de paquete mayor información multimedia [29].

Su uso actualmente se encuentra en aumento, existiendo multitud de emisoras de radio y televisión que utilizan el *streaming* para compartir sus contenidos, como por ejemplo las televisiones y radios más importantes de nuestro país. Aunque encontramos otros ejemplos como las videoconferencias, con usos que van desde el empresarial hasta la tele educación; o los videoclubs

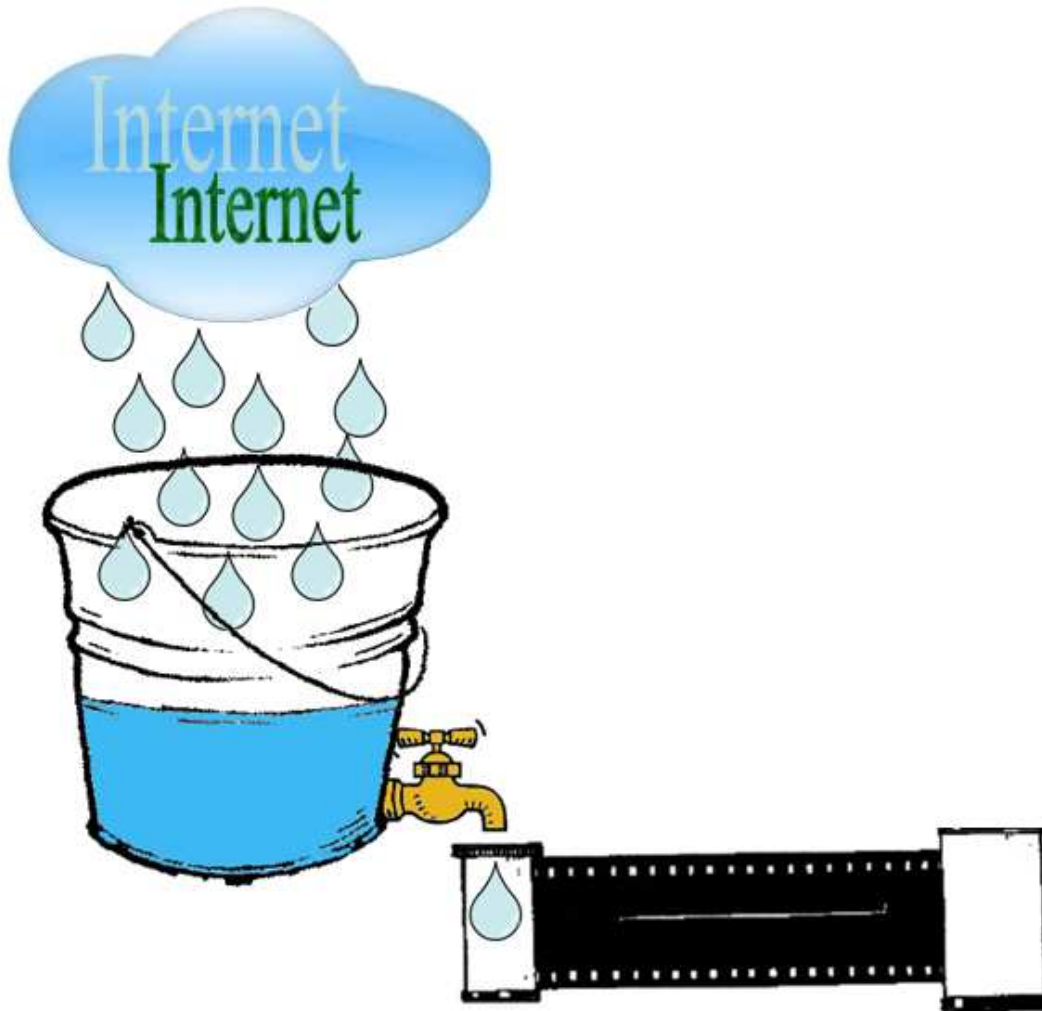


Figura 2.6: Ejemplo de almacenamiento mediante buffer

online, que ofrecen contenidos multimedia a sus clientes, ya sea de manera gratuita o a través del pago por los contenidos [29].

Trataremos con mayor profundidad dos aspectos importantes en este tipo de transmisión de información, por un lado los protocolos que intervienen, y por otro los tipos de envío de información más importante que encontramos.

CAPÍTULO 2. ESTADO DEL ARTE

■ ■ ■ 2.3.0.4 Protocolos

■ Protocolos de Transporte

Los protocolos de transporte que se utilizan en este tipo de transmisión son:

Transport Control Protocol (TCP)

TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión, siendo uno de los principales protocolos, dando la posibilidad de administración al protocolo IP [30].

Las principales características del protocolo TCP son las siguientes [30]:

- Permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- Permite el monitoreo del flujo de los datos y así evita la saturación de la red.
- Permite que los datos se formen en segmentos de longitud variada para entregarlos al protocolo IP.
- Permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, aplicaciones) en la misma línea pueda circular simultáneamente.
- Permite comenzar y finalizar la comunicación amablemente.

User Datagram Protocol (UDP)

Protocolo de transporte no orientado a conexión, permite el envío de tráfico IP sin necesidad de establecer una conexión a través, contando con una sencilla cabecera donde aparecen los puertos utilizados para el envío. Es utilizado para protocolos como DNS [30].

Por todo lo anterior diremos que no proporciona confirmación ni control de flujo en la conexión, por lo que los paquetes pueden llegar al destino en un orden diferente del que se envió; y tampoco se sabe si ha llegado correctamente cada paquete, ya que no hay confirmación de entrega o recepción [30].

Real-time Transport Protocol (RTP)

La función básica de RTP es multiplexar varios flujos de datos en tiempo real en un sólo flujo de paquetes UDP. Este flujo puede ser enviado a un único destino o a varios.

Al utilizar UDP sus paquetes no son tratados de manera especial por los nodos de la red, a menos que se habilite alguna característica de calidad de servicio [30].

A cada paquete enviado por un flujo UDP se le da un número superior al anterior, para que de esta forma pueda conocerse en el destino la numeración adecuada, adoptándose medidas en caso de faltar algún paquete, siendo lo más habitual la interpolación de la muestra perdida. En la comunicación en tiempo real no se recomienda la retransmisión en el envío de datos debido a que puede provocar mayores retardos [30].

- Protocolos de Sesión

Real-time Transport Streaming Protocol (RTSP)

Protocolo utilizado en sistemas *streaming*, que permite al cliente controlar de manera remota el acceso a los archivos del servidor de *streaming* por medio de comandos sencillos, como pueden ser reproducir o pausar. Para acceder a los archivos se utiliza RTP [30]

Real-time Transport Control Protocol (RTCP)

El protocolo RTCP se basa en transmisiones periódicas de paquetes de control que realizan todos los participantes de la sesión [30].

Es un protocolo de control para el flujo RTP, que permite transmitir información básica sobre los participantes de la sesión y la calidad de servicio [30].

ReSerVation Protocol (RSVP)

Utilizado en la arquitectura de calidad de servicios *IntServ* (Servicios Integrados) para la reserva de recursos de una red, es considerado un protocolo de control de reserva de canales o rutas a través de las cuales serán transmitidos paquetes a los que se les proporciona una calidad de servicio específica [30].

■ ■ ■ 2.3.0.5 Tipos de *Streaming*

- *Streaming* en Tiempo Real

Nos encontramos este tipo de *streaming* cuando la grabación de contenidos y la reproducción de los mismos se realizan de manera sincronizada, solamente variando el tiempo entre ambos debido al retraso que se introduce en el servidor por las labores de procesamiento

CAPÍTULO 2. ESTADO DEL ARTE

de la información, codificación de la misma y puesta en la web para su difusión, un ejemplo lo encontramos en la transmisión de eventos en directo [29].

- Vídeo bajo Demanda (*Video On Demand* (VOD))

Si los contenidos fueron grabados previamente, junto con su posterior postprocesamiento en el servidor, y después son ofrecidos a los usuarios. Aquí los usuarios realizarán una petición al servidor de la información que desean, y tendrán la capacidad de comenzar la reproducción, pausarla, pararla, avanzar o retroceder por el contenido, un ejemplo se encuentra en la mayoría de los contenidos que YouTube [31] pone a nuestra disposición [29].

2.4 Dispositivos Móviles

El teléfono móvil ha producido un cambio muy importante en la sociedad, por un lado se ha convertido en algo indispensable dentro de nuestra forma de vida, viajando junto a las llaves y la documentación, pero también podemos observar como en unos pocos años el número de líneas de telefonía móvil en nuestro país ha superado al de personas [32].

Por otro lado, los gustos y necesidades de los usuarios han ido variando tal y como lo ha hecho la tecnología, dispositivos cada vez más pequeños, pantalla a color, cámara de fotos o *bluetooth* por poner algún ejemplo; y por ese motivo cada vez se busca un equipo que defina mejor las necesidades y gustos actuales de los usuarios.

El último paso ha sido la aparición de los *Smartphones*, que mezclan los usos de un teléfono móvil con las utilidades de un ordenador de sobremesa, permitiendo nuevas tareas que van desde leer el correo hasta dar la posibilidad de entrar en redes sociales. Todo ello es debido a que estos dispositivos llevan instalado un SO que les permite realizar todas estas tareas de una forma rápida y sencilla.

A continuación, realizaremos una breve descripción de los SO para dispositivos móviles y *smartphones* más importantes que se encuentran en el mercado a día de hoy, incluyendo una comparativa de las características de cada uno de ellos.

2.4.1 Symbian

Los orígenes de este sistema operativo provienen de su antepasado EPOC32, de la empresa *Psion*, que era utilizado mayoritariamente en *Personal Digital Assistant* (PDA), y que aún hoy podemos ver que existe una librería del SO llamada *epoc32* [33].

Más tarde, en 1998 se funda *Symbian Ltd*, sociedad formada por *Ericsson*, *Nokia*, *Motorola* y *Psion*, para intentar explotar la convergencia entre teléfonos móviles y PDA, sin embargo pasados unos años *Motorola* vendió su parte de *Symbian* a *Nokia* [33].

En 2008 pasó a manos de *Nokia*, el cual fundó la Fundación *Symbian*, con el objetivo de ofrecer un SO bajo una plataforma abierta, pasando de este modo los empleados de *Symbian* a ser trabajadores de *Nokia*. Actualmente la Fundación *Symbian* está formada por varios miembros como son *HP*, *MySpace*, *SanDisk* o *Qualcomm*, aparte de *Nokia* [33].

La cuota de mercado de *Symbian* se ha visto reducida en los últimos años, como puede verse en la Figura 2.7, adquiriendo mayor protagonismo *iPhone*, *RIM* y *Android*. El número de dispositivos móviles que utilizan *Symbian* han pasado de ser casi el 70 % a estar algo por encima del 40 % del total de terminales [33].

Aunque a finales de 2010 se espera la llegada de *Symbian*⁴, actualmente se está trabajando en la versión *Symbian*³ que vio la luz el pasado mes de Febrero, presentando en esta versión del SO la integración de Qt 4.6 para el desarrollo de interfaces gráficas de aplicaciones [33].

Qt es una biblioteca de herramientas multiplataforma para desarrollar interfaces gráficas, aunque también puede ser usada para programas sin interfaz, es originario de la empresa *Trolltech*, sin embargo, está en manos de *Nokia* desde 2008. En la versión *Symbian*⁴ se espera que el desarrollo de aplicaciones se haga usando Qt, que es utilizado para desarrollar software de la talla de *Skype* o *Lex* (Editor gráfico de L^AT_EX) [33].

Las principales novedades de este SO son: una nueva interfaz gráfica totalmente rediseñada y adaptada; una reestructuración de la arquitectura de red; orientada a la reproducción de audio y vídeo; conectividad HDMI para poder visualizar vídeos en alta resolución conectando el móvil al televisor; y avances que hacen más sencillo el uso [33].

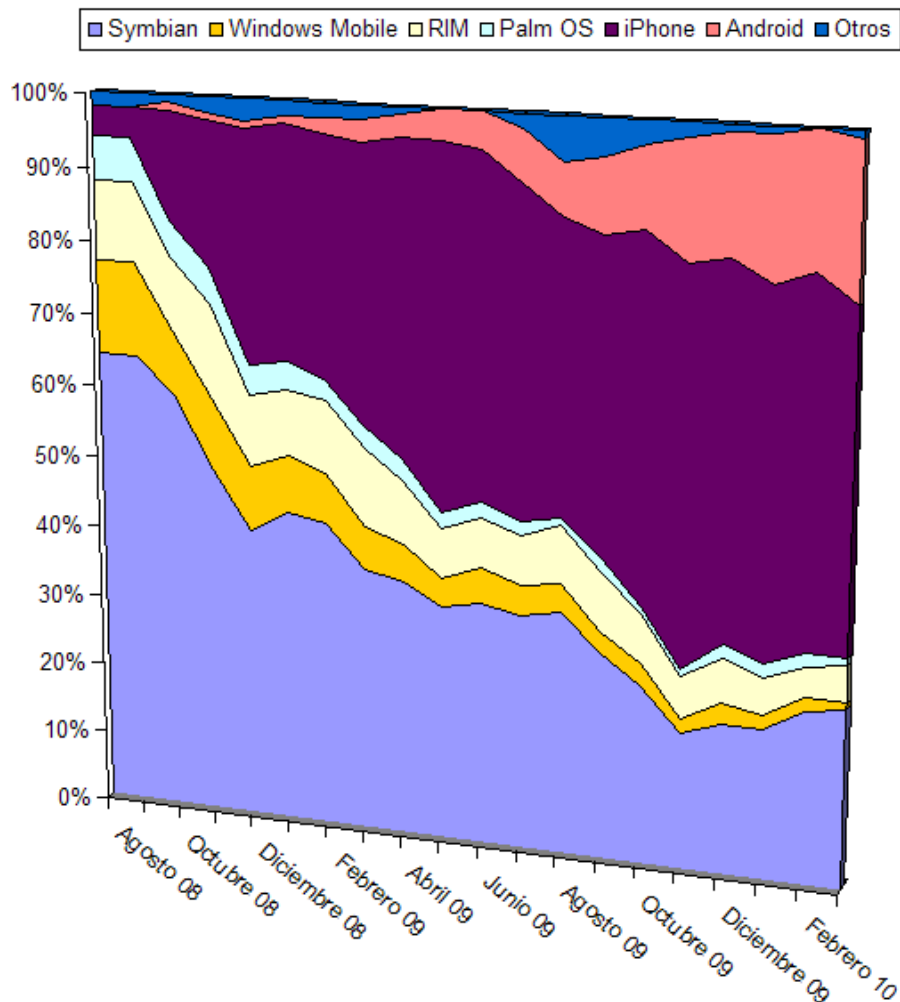


Figura 2.7: Cuota de mercado de SO móviles [3]

■ ■ 2.4.2 Windows Mobile

La línea de negocio de la empresa de *Microsoft* dedicada a los dispositivos reducidos ha sido conocida como *Windows Mobile*, trabajando actualmente en *Windows Phone*, en concreto en *Windows Phone 7*. Fue a principios de los años '90 cuando *Microsoft* decidió sacar un sistema operativo para dar respuesta a la demanda emergente del mercado en este tipo de dispositivos, y lo hizo bajo el nombre de Pegasus, más conocido por *Windows CE* [34].

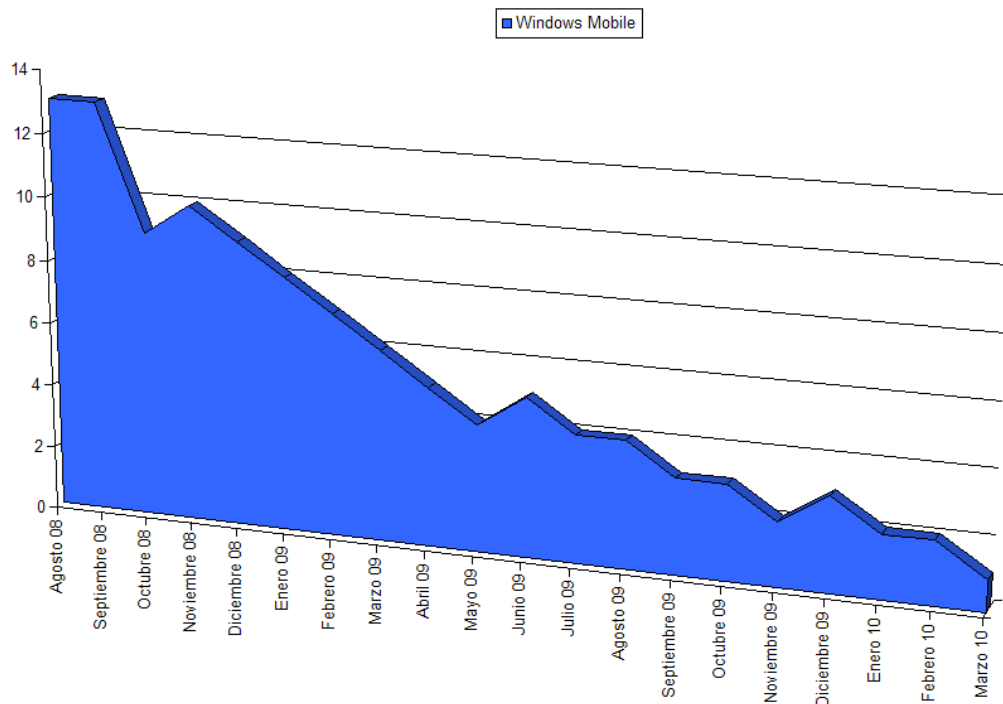


Figura 2.8: Cuota de mercado de Windows Mobile [3]

Por otra parte, aunque este SO no ha gozado de una cuota de mercado elevada, desde hace unos años lleva una línea descendente en cuanto a cuota de mercado se refiere (ver figura 2.8), paralela a la que han sufrido *Palm OS* o RIM (*Blackberry*), y contraria al crecimiento que han experimentado las plataformas *iPhone* y *Android*, que han visto aumentada su cuota en estos dos últimos años [34].

En febrero de este año se anunció la salida al mercado de la versión *Windows Phone 7*, nueva plataforma que está llamada a aumentar el posicionamiento de *Microsoft* en el mercado de *smartphones*. Según comentaba Steve Ballmer, consejero delegado de *Microsoft*, "en un mercado lleno de gente con un montón de teléfonos que tienen el mismo aspecto y hacen las mismas cosas, *Windows Phone* Serie 7 marca un punto de inflexión hacia los teléfonos que reflejan realmente

CAPÍTULO 2. ESTADO DEL ARTE

la velocidad de la vida de las personas" [35].

Durante la conferencia MIX2010 de *Microsoft*, que tuvo lugar el mes de marzo del año 2010, la firma anunció las bondades del nuevo SO. Podemos destacar que los dispositivos que lo usen tendrán una pantalla capacitiva de 800x480 píxeles, cámara de fotos no inferior a 5 megapíxeles, 256 MB de memoria RAM, junto a 8 GB de memoria flash, procesador *ARMv7 Cortex/Scorpion*, o compatibilidad con *DirectX 9*; características que buscan dar una experiencia de uso elevada, unida a su nuevo diseño con actualización instantánea [34].

Sin embargo, la multitarea fue uno de los puntos más comentados, debido a que *Windows Phone 7* no soporta multitarea. Sólo las aplicaciones básicas podrán ejecutarse en segundo plano, argumentando desde la firma motivos de ahorro de batería. Otra falta no tan acusada fue la opción de copiar y pegar que tampoco permite este SO [34].

■ ■ 2.4.3 Blackberry

El primer equipo de esta compañía fue un dispositivo buscapersonas que hizo su aparición a finales del siglo pasado de la mano de *Research in Motion Limited* (RIM), compañía canadiense. Desde un principio instauró entre sus principales aplicaciones un organizador, correo, calendario y acceso inalámbrico, puntos que se convirtieron en pilares fundamentales de esta plataforma [36].

La historia de *Blackberry* va unida a un aumento de las ventas de terminales, y por tanto, de servicios ofrecidos por RIM a compañías telefónicas, y estas a los usuarios finales, base de su modelo de negocio. Así pues, pasaron de tener 3 millones de terminales en mayo de 2005, hasta los casi 30 millones en el año 2009, ocupando más de un 40 % de la cuota de mercado de los smartphones en Estados Unidos [36].

Actualmente cuenta con la versión del SO 6.0 de RIM [36], que trae consigo características ya famosas de *Blackberry* como puede ser el soporte multitarea, aplicaciones de correo, calendario y agenda, o la conectividad permanente a internet, y otras nuevas como el soporte a una navegación más gestual e intuitiva con ayuda del *trackpad*, que sustituye a la rueda de desplazamiento o *trackball* [36].

2.4. DISPOSITIVOS MÓVILES

Ofrece soporte a *Mobile Information Device Profile* (MIDP) y *Wireless Application Protocol* (WAP), así como aplicaciones desarrolladas en Java, además, los desarrolladores pueden hacer uso del *Application Programming Interface* (API) que *Blackberry* pone a su disposición. Un aspecto a destacar es que para hacer uso de determinadas características de los terminales las aplicaciones han de estar firmadas, yendo asociadas a un número de usuario de desarrollador [37].

Entre sus características principales encontramos la mayoría del resto de *smartphones*, aunque podemos destacar como símbolo de *Blackberry* el teclado *qwerty* que han desarrollado en la mayoría de sus modelos, implementando una letra por cada tecla [36].

■ ■ 2.4.4 Palm OS

En 1992 Jeff Hawkins funda la compañía, que empezó a comercializar una PDA llamada *Zoomer*, con SO de *Geoworks* y fabricada por *Casio*, la cual resultó un fracaso que hizo que la compañía sobreviviera gracias al desarrollo de software para terceros. En 1995 pasó a manos de *U.S. Robotics*, qué más tarde compraría 3Com (1997), lo que propició la salida de los fundadores para fundar *Handspring* al año de entrar 3Com en la dirección [38].

3Com transformo la compañía de subsidiaria a independiente, llegando al *Nasdaq* en el año 2000 perdiendo cerca del 90 % del valor de las acciones en un año. En el año 2002 Palm se dividió en dos empresas, PalmSource encargada del SO y palmOne que se encargaría del Hardware, uniéndose a Handspring [38].

En el año 2005, *palmOne* adquirió *PalmSource* cambiando el nombre y la marca de nuevo a *Palm*, pero antes de acabar ese año, *Acces* adquirió *PalmSource* por 324 millones de dolares, sacando al mercado el año 2006 el dispositivo *Palm Treo 700w*, con el SO *Windows Mobile* [38].

Pero toda esta historia turbulenta aún no había tocado a su fin, fue en 2006 cuando *Palm* pagó 44 millones de dolares a la empresa *Acces*, para adquirir derechos de código de *PalmOS* y así poder trabajar tanto en el software como en el *hardware* de sus *smartphones*. En junio del año 2007 *Palm* se alió con *Elevation Partners*, quién pasó a poseer el 25 % de la compañía [38].

Durante el *Consumer Electronic Show* (CES) de Las Vegas, en enero de 2009, *Palm* anunció

CAPÍTULO 2. ESTADO DEL ARTE

un nuevo sistema operativo para sus dispositivos, *Palm webOS*, junto a la salida al mercado de un nuevo dispositivo *Palm Pre* [38].

En marzo de 2010 *Palm* declaró las pérdidas del año anterior, declarándolo como decepcionante, y surgiendo de esa forma los primeros rumores de venta de la compañía, que se materializaría el 28 de abril de 2010 con la compra por parte de HP por la cantidad de 1.200 millones de dolares [38].

WebOS es un SO multitarea basado en *Linux*, que ha sido diseñado a través de una interfaz de usuario con pantalla táctil y hace uso de tecnologías de programación web como *Extensible HyperText Markup Language* (XHTML), *JavaScript* y *CSS*, pensado para que los desarrolladores no tengan que aprender un nuevo lenguaje de programación [38].

Quizá uno de los aspectos más importantes es la inclusión de *Synergy* (gestor de información dispersa para mostrar dicha información como un sólo dato), por ejemplo, si tuviésemos el mismo contacto en el correo, la agenda y la popular red social *Facebook* los detectaría como la misma persona, actualizando la información de manera automática si ocurriera algún cambio [38].

También destacar que algunos expertos achacan como posible mejora la ampliación del ecosistema de aplicaciones para este dispositivo, pero sobre todo las vulnerabilidades de seguridad demostradas por *Intrepidus Group* de la versión 1.3.5 [39].

■ ■ 2.4.5 iPhone

El día 9 de enero de 2007 Steve Jobs publicó una *keynote* dentro de la *MacWorld* de ese mes presentando un nuevo equipo que, según Jobs, iba a revolucionar la telefonía, aunque fue en 2004 cuando comenzó a gestarse este dispositivo, a raíz de la creciente demanda de nuevos equipos móviles multimedia que ponían en peligro el modelo de negocio de *Ipod* e *iTunes* [40].

Fruto de ello nació una unión entre *Motorola*, *Cingular* (actual AT&T) y *Apple* que dio lugar a *ROKR* en septiembre de 2005, que supuso un fracaso en las expectativas de venta y de negocio, al no poder almacenar más de 1000 canciones y no ser posible la descarga directa desde *iTunes* [40].

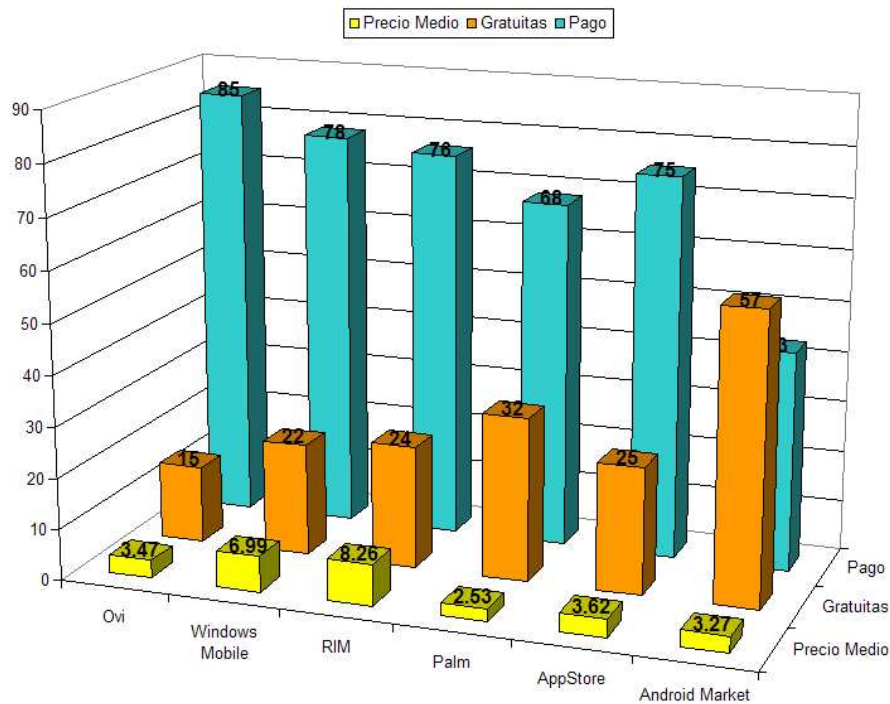


Figura 2.9: Precio de las Aplicaciones para Smartphones [3]

Así pues Apple puso en marcha el proyecto Purple2, que junto a la pantalla multitáctil que habían conseguido desarrollar los ingenieros de la marca, los procesadores más veloces, y la filosofía que lograron plasmar en el SO es lo que se conoce como *iPhone*, el cual en poco tiempo consiguió una cuota de mercado elevada, estando instalado en el 50 % de los equipos en el mercado de Smartphones [40].

El SO se comercializa en el *iPod Touch*, *iPhone*, en todos sus modelos, e *iPad*, trabajando actualmente en la versión 4.0 del mismo, que experimentó su salida en verano de 2010 [40]. El *iPhone OS* está basado en una variante del *kernel* de *Mac OS X* que incluye componentes de software y *hardware* como Animación *Core* o *PowerVR MBX*, está compuesto por 4 capas de abstracción: la capa del núcleo del sistema operativo, la capa de Servicios Principales, la capa

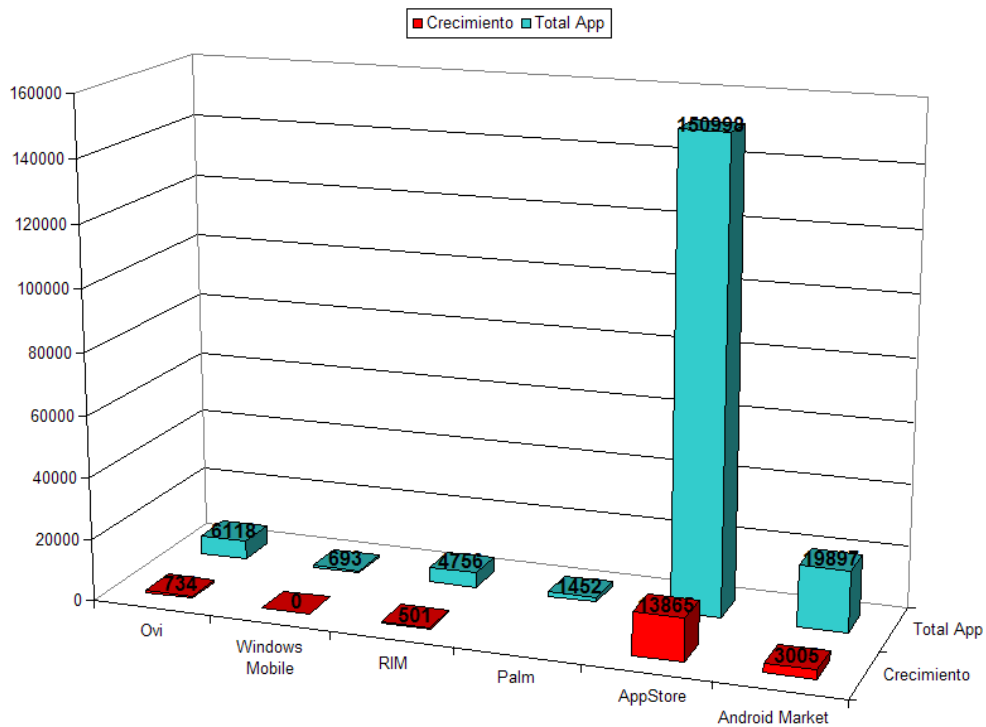


Figura 2.10: Volumen de Aplicaciones para Smartphones [3]

de Medios de comunicación y la capa de *Cocoa Touch*.

Entre los principales aspectos de la versión 4.0 podemos destacar la multitarea que no compromete la batería, basada en servicios diferentes para que los desarrolladores puedan adaptar sus aplicaciones:

- Background Audio
- Voice over IP
- Background Location
- Push and Local Notification

- Task Completion
- Fast Application Switch

Además de la novedad esperada de la multitarea, encontramos entre las más importantes mejoras del sistema operativo respecto a versiones anteriores soporte a *iBooks*, publicidad en las aplicaciones, unificación de cuentas de correo electrónico, agrupación del escritorio en carpetas, y una nueva plataforma de juegos en la línea de *Xbox live* [40].

Además, destacamos la tienda de aplicaciones *AppStore*, uno de los puntos fuertes de este SO, ofrece a los usuarios una gran colección de aplicaciones tanto de pago como gratuitas, vease Figura 2.9 y Figura 2.10 [40].

■ ■ 2.4.6 Android

Una pequeña compañía de Palo Alto, California, llamada *Android Inc.* fue comprada por *Google* en el verano del 2005, trascendiendo rumores sobre la posibilidad de que esta última compañía fuera a sacar al mercado un equipo gratuito, enfocado a sacar ganancias por publicidad en el terminal. Dos años más tarde, en Noviembre de 2007, se fundó la *Open Handset Alliance*, un conglomerado de casi 50 compañías de *hardware*, software y telecomunicaciones con el objetivo de promocionar los estándares abiertos para dispositivos móviles [41].

Al poco tiempo se lanzó el SDK de *Android*, libre y gratuita para los programadores de aplicaciones, el cual incluye un plugin para el entorno de programación Eclipse, y un emulador para poder realizar pruebas con el software programado [41].

Dalvik2.11 es la máquina virtual de *Android* que se encuentra por encima del *kernel Linux* del SO, su función es interpretar el *Bytecode* de Java y transformarlo a *Bytecode* entendible por *Dalvik*, migrando los archivos *class* a formato *dex*, y es por ese motivo que la mayor parte de las aplicaciones en *Android* son programadas en Java, aunque no es la única manera, ya que está disponible un entorno de programación a nivel de *kernel*, lo que se conoce como *Native Development Kit* (NDK), que permite la integración de librerías C/C++, utilizado para sacar mayor rendimiento gráfico en la programación [41].

CAPÍTULO 2. ESTADO DEL ARTE

Desde que hiciera su aparición en el mercado, *Android* ha tomando un protagonismo que ha ido en aumento, haciéndose constar en el estudio elaborado por *NPD Group* sobre el número de terminales vendidos en Estados Unidos, superando a *iPhone* y ocupando el segundo puesto tras *Blackberry*, algo que han puesto en duda desde *iPhone* [42].

Desde que en agosto de 2008 apareciera la versión 0.9 del SDK de *Android* hasta la versión *Froyo* han salido diferentes versiones de este SO [41]:

- Android 1.5 Cupcake:
Soporte para teclado en pantalla, widgets y grabación de vídeo.
- Android 1.6 Donut:
Soporte para múltiples resoluciones, caja de búsquedas, VPN, 802.1X y Android Market
- Android 2.1 Eclair:
Soporte para multitouch, Google Maps navigation, sincronización de múltiples cuentas y mejora en el Browser.
- Android 2.2 Froyo:
Actualmente se trabaja en la versión Froyo, que hizo su aparición en la reciente Google I/O del mes de mayo de 2010 Soporte para 802.11n, Radio FM, Flash, API gráfica OpenGL, tethering por USB, hotspot WiFi y mejora de rendimiento próxima al 450 %.

Android busca reunir bajo la misma plataforma todas las herramientas necesarias para el programador, con el fin de aprovechar al máximo las funciones que ofrece un dispositivo reducido, como puede ser un teléfono móvil [41].

Podemos observar la arquitectura de *Android* en la Figura 2.11, que se compone de los siguientes componentes principales [41]:

1. *Kernel de Linux*

Android está basado en la versión 2.6 del *kernel* de *Linux*, realizando labores de abstracción entre *hardware* y software, funciones de seguridad, o gestión de memoria entre otros.

2.4. DISPOSITIVOS MÓVILES

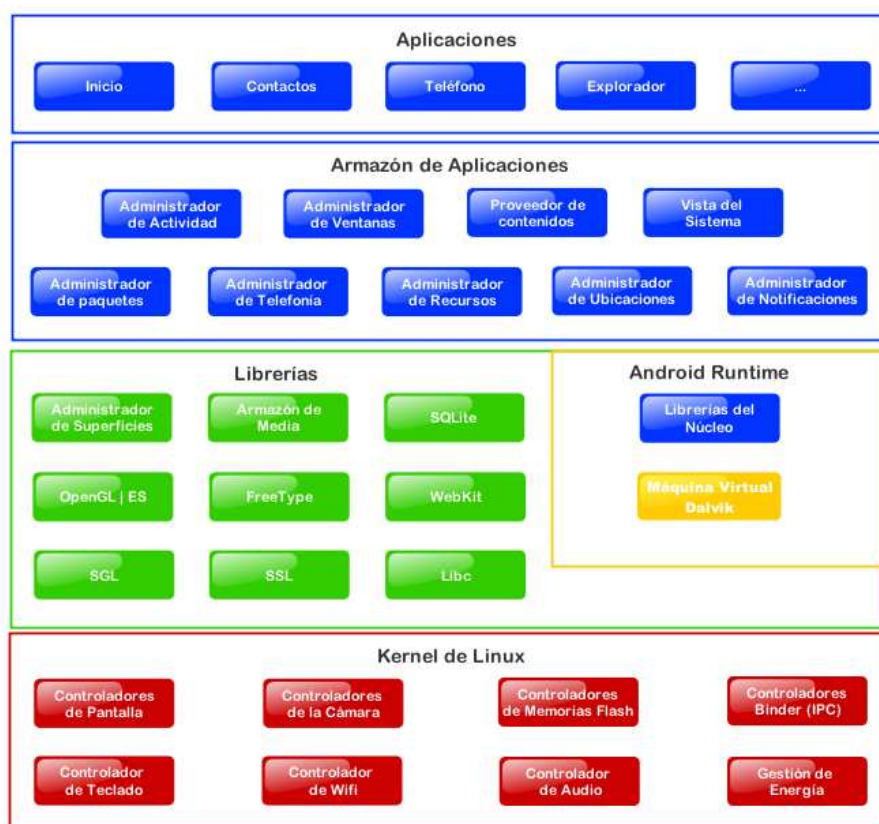


Figura 2.11: Arquitectura Android [4]

2. *Android Runtime*

El uso de *Dalvik* se traduce en una instancia de la máquina virtual por cada proceso ejecutado.

3. Librerías

Android incluye un conjunto de librerías que utilizan componentes del sistema, y que están disponibles para el programador a través de las aplicaciones marco.

4. Framework

Al programar se tiene acceso a los marcos de las aplicaciones básicas, permitiendo la reutilización de componentes, debido a que cualquier aplicación puede dar a conocer su uso, permitiendo a otras aplicaciones hacer uso de ellas.

CAPÍTULO 2. ESTADO DEL ARTE

5. Aplicaciones

Con *Android* se entregan instalados varios programas básicos para un *smartphone*, como puede ser calendario, contactos, o mapas, aunque podremos actualizar las mismas a través de *Android Market*.

■ ■ 2.4.7 Otros SO para dispositivos reducidos

Si bien los SO antes descritos son los que actualmente poseen mayor importancia, pero existen otros que están llamados a competir con los anteriores en lo que a cuota de mercado se refiere, con lo que haremos una breve descripción de los mismos.

■ ■ ■ 2.4.7.1 bada

El nombre de *bada* es la traducción de océano del coreano y aparece de la mano de *Samsung*, pretende ser una plataforma bajo la cual los *smartphones* puedan acercarse a un mayor sector de la población. La idea es lograr una experiencia de uso confortable en dispositivos tan avanzados como puede ser *Samsung Wave*, y a su vez equiparlos en equipos con una capacidad de procesamiento menor [43].

Aunque *Samsung* es el propietario del SO, proyecto que cuenta con una antigüedad de unos 10 años, no se centrará exclusivamente en él, al menos durante este año 2010 [44]. La idea de este SO gira en torno a una SDK libre para los programadores, asegurándose así parte de la oferta a la que se puede acceder desde las tiendas de aplicaciones para teléfonos inteligentes [43].

■ ■ ■ 2.4.7.2 MeeGo

Durante el pasado *Mobile World Congress* que tuvo lugar en Barcelona durante el mes de Febrero de 2010, *Intel* y *Nokia* anunciaron la aparición de *MeeGo*, fruto de la fusión de los SO *Moblin* y *Maemo* de las respectivas marcas [45].

MeeGo está pensado para funcionar en teléfonos móviles, *netbooks*, vehículos o televisores. Se

apoya en una distribución de *Linux* que soporta *ARM* e *Intel Atom*, y al igual que *Symbian*, usará *Qt* para desarrollar la interfaz gráfica, teniendo el beneplácito de *The Linux Foundation* [45].

■ ■ ■ 2.4.7.3 LiMo

Linux Mobile dan nombre a las siglas de esta fundación que data del 2007 cuando *Motorola*, *NEC*, *NTT DoCoMo*, *Panasonic*, *Samsung* y *Vodafone* decidieron unirse para sacar al mercado un SO completamente gratuito [46].

La idea sobre la que se basa *LiMo* es realizar una plataforma completamente independiente del *hardware*, donde cada aplicación pueda ser utilizada sin importar el soporte físico en el que se encuentra instalado el SO, bien sea un *smartphone* o un vehículo [46].

■ ■ ■ 2.4.7.4 Java Micro Edition

Aunque esta plataforma de *Sun* no es un SO por si mismo, el lugar que ocupa en el mercado de desarrollo de telefonía móvil es de tal envergadura que se hace imprescindible dedicarle unas líneas [47].

Antes conocido como *Java To Micro Edition* (J2ME), es una especificación de un subconjunto de Java que provee una colección de API de desarrollo para dispositivos con recursos limitados, como puedan ser electrodomésticos, PDA o teléfonos móviles, aunque la mayor introducción se debe a la programación de juegos para telefonía [47].

■ ■ 2.4.8 Comparativa de Sistemas Operativos

A continuación haremos un análisis de las principales prestaciones de estos SO, comparando cada aspecto para las principales plataformas.

SO	Symbian	Windows Mobile	RIM/Blackberry	Palm OS	iPhone	Android
Kernel	Symbian	Windows CE	Propietario	Linux	Mac OS	Linux
Multitarea	Si	No	Si	Si	No	Si
Antigüedad	Madura	Madura	Madura	Joven	Adolescente	Joven
Adaptabilidad	Excelente	Excelente	Buena	Excelente	Mala	Excelente

Tabla 2.1: Aspectos Básicos de los SO para móviles [6]

2.4.8.1 Aspectos Básicos

Una parte básica de un *smartphone* es su SO, tanto *Android* como *Palm* se basan en *Linux*, *iPhone OS* lo hace en OS X, variante de *Unix*, *Blackberry* tiene un SO propietario, y *Windows Phone* por su parte se basa en *Windows CE*. La diferencia de usar un sistema propietario o no se resume en la comunidad de usuarios que conocen y ayudan a la mejora de los mismos, debiendo dedicar más tiempo los sistemas propietarios a la detección de fallos de seguridad puesto que su comunidad se reduce a la propia empresa, podemos ampliar información en la Tabla 2.1.

La adaptabilidad es otro punto importante, puesto que nos va a permitir utilizar ese SO en diversidad de equipos *hardware*, como puedan ser *netbooks*, sistemas empotrados o automóviles. De todas las plataformas vistas, la que tiene mayor adaptabilidad es Android, a pesar de su corta vida, ampliándose su uso a otros equipos que no sólo son *smartphones*.

SO	Symbian	Windows Mobile	RIM/Blackberry	Palm OS	iPhone	Android
Intuitividad	Baja	Alta	Alta	Alta	Alta	Alta
Gestos	Meida	Sencilla	Sencilla	Sencilla	Sencilla	Sencilla
Pantalla	Cap./Res.	Resistiva	Capacitiva	Capacitiva	Capacitiva	Capacitiva
Instalaciones	Media	Sencilla	Sencilla	Sencilla	Sencilla	Sencilla
Notificación	PopUp	PopUp,Bandeja	PopUp	Bandeja	PopUp	Bandeja
Contactos	Exchange	Exchange	BES, BIS	Synergy	Mac OS	Google
Multimedia	Ovi	Media Player	iTunes	Amazon	iTunes	Amazon
Actualización	Teth.,OTA	Teth.,OTA	Teth.,OTA	Sin Datos	Teth.	OTA
Navegador	Webkit	Explorer	Propietario	Webkit	Webkit	Webkit

Tabla 2.2: Interacción con Usuario [6]

2.4.8.2 Interacción con el Usuario

Sin lugar a dudas el *kernel* es uno de los puntos más importantes del equipo, sin embargo para un usuario es más importante la sencillez y velocidad de la interfaz de usuario. En esa línea de experiencia amigable trabajan los SO, habiendo marcado un hito la facilidad de uso que trajo consigo *iPhone* y la pantalla multitáctil, podemos ver una comparativa de las aplicaciones principales que los usuarios utilizan en la Tabla 2.2.

El grado de personalización también es importante, así vemos como la nueva versión de *iPhone* trae, entre otras cosas, la inclusión de carpetas para organizar las aplicaciones, pero sigue sin dejar cambiar el tamaño de fuente (una de las pocas modificaciones que permite *Blackberry*), algo que en *Android* y *Windows Mobile* aparece como opción habitual desde su lanzamiento. En esta línea también está el hecho de actualizar el terminal, bien por vía *Over The Air* (OTA), o

CAPÍTULO 2. ESTADO DEL ARTE

a través del modo *Tethering*¹. del teléfono, resultando completamente transparente la primera opción de las dos.

La experiencia del usuario también estará modelada por el teclado del equipo, pudiendo darse el caso de que sea físico, con la problemática del reducido tamaño de las teclas pero sin reducir la pantalla por el hecho de introducir texto tal y como ocurren en los teclados virtuales.

2.4.8.3 Desarrollo de Aplicaciones

SO	Symbian	Windows Mobile	RIM/Blackberry	Palm OS	iPhone	Android
Disponibilidad SDK	Si	Si	Si	Si	Si	Si
Tienda Aplicaciones	Prox.	Si	Prox.	Si	Si	Si
Disponibilidad Aplicaciones	Media	Alta	Media	Baja	Alta	Alta
Aplicaciones Nativas	Si	Si	No	No	Si	Si
Admon. Aplicaciones	Buena	Buena	Buena	Excelente	Excelente	Excelente

Tabla 2.3: Desarrollo de Aplicaciones [6]

Dar la posibilidad a los usuarios de poder incluir nuevas funcionalidades es esencial para adquirir una importancia mayor en esta lucha por los SO de los *smartphones*, y por ello es que la mayoría de las plataformas ponen a disposición de los programadores un SDK (véase Tabla 2.3), en algunos casos más accesible que en otros, para que puedan desarrollar aplicaciones.

Junto a ello se encuentra la tienda de aplicaciones en cada plataforma, que tal y como vimos en la Figura 2.7, lleva ventaja *iPhone* al resto.

¹Se conoce *tethering* (traducido del Inglés participio del verbo atar) como aquel proceso por el cual un dispositivo móvil con conexión a Internet actúa como pasarela para brindar acceso inalámbrico a la red a otros dispositivos.

2.5 Sistemas de Audioguías

Una audioguía es un equipo de reducido tamaño que es utilizado en lugares de interés turístico para ofrecer al turista información didáctica del entorno en el que se encuentran. Aunque su uso se extiende a otros lugares, habitualmente las encontramos en museos y galerías de arte, permitiendo al usuario recibir información en forma de audio a partir de la introducción de un código, tal y como lo haría una persona física, existiendo además la posibilidad de implementarlo en diferentes idiomas [5].

El modo de uso de la misma podemos verlo en la Figura 2.12, que consiste en dos partes, por un lado las labores que la institución turística ha de realizar, que se basan en la carga de las baterías y el mantenimiento de los equipos, y por otro dotar a los equipos de los archivos adecuados, tanto en idioma como en contenido, para que los usuarios tengan una experiencia de uso adecuada. Los usuarios por su parte deberán identificar los códigos que le indicarán al equipo que archivo ha de reproducir, ya sean esos códigos de forma alfanumérica o algún otro medio, como puede ser un código de barras, para posteriormente disponer de la reproducción que demandan [5].

■ ■ 2.5.1 Arquitectura del Sistema

La arquitectura de este tipo de sistemas es sencilla, a través de la introducción de un código el sistema selecciona el audio a reproducir, disponiendo el usuario de algunos controles como puede ser la introducción de datos (por medio de un código que se teclee en el equipo, o a través de códigos de barra que es capaz de interpretar el dispositivo, por ejemplo), variar el idioma, manejar el volumen o los típicos botones de reproducción, son algunos ejemplos de las posibilidades del sistema [5].

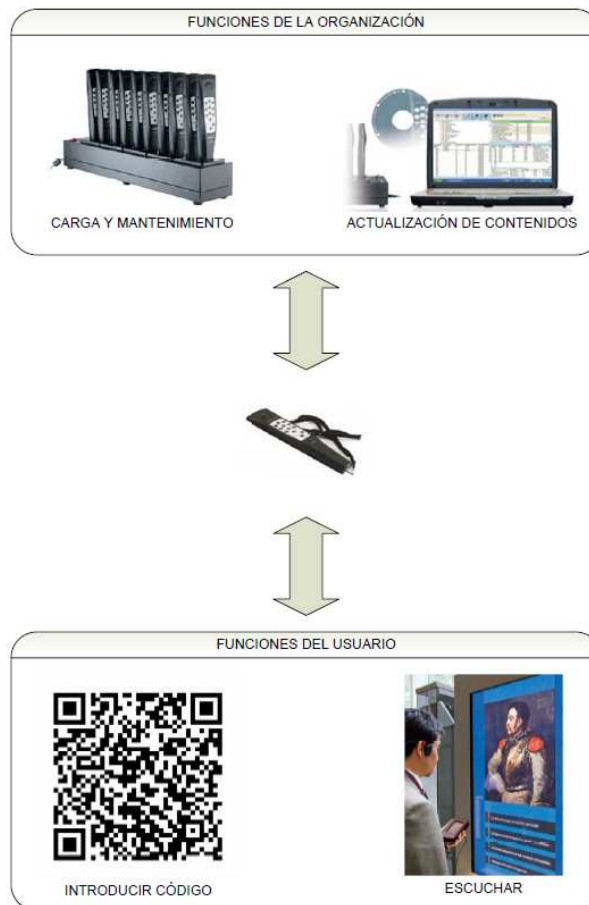


Figura 2.12: Actores de un sistema de audioguía [5]

■ ■ 2.5.2 Sistemas Actuales

En la actualidad estos sistemas no se basan únicamente en el modelo tradicional de audioguía, sino que van más allá, permitiendo diferentes tipos de contenidos, y pudiendo llegar a una cantidad de usuarios mayor [5].

Por ello podemos agrupar estas guías en diferentes grupos: tradicionales, multimedia y en dispositivos externos, que repasaremos a continuación.

2.5. SISTEMAS DE AUDIOGUÍAS

2.5.2.1 Tradicionales

Como su nombre indica, estos equipos llevan a cabo la labor de ofrecer a los usuarios archivos de audio referenciados a códigos que son introducidos por los primeros para acceder a la información [5].

La numeración no se deja al azar, sino que lleva a cabo la típica visita que nos haría un guía turístico, pero sin la necesidad física de este, y permitiendo un orden aleatorio si el usuario lo cree conveniente [5].

2.5.2.2 Multimedia

El concepto tradicional de audioguía se basa en la transmisión de información por medio del audio, pero existen equipos que aumentan esa experiencia ofreciendo, además, la posibilidad de reproducir un archivo de vídeo, o un texto [5]. Las signoguías (Figura 2.13) se están convirtiendo



Figura 2.13: Ejemplo de signoguía para personas con problemas auditivos [5]

en una plataforma extraordinaria para acercar este tipo de aplicaciones a las personas con discapacidades físicas, en este caso con atrofia auditiva [5].

Una signoguía se basa en la reproducción de un archivo de vídeo en el que se muestra a una persona traductora del lenguaje de signos para personas con problemas auditivos, además suelen ir acompañadas de un subtítulo [5].

CAPÍTULO 2. ESTADO DEL ARTE

■ ■ ■ 2.5.2.3 Equipo externo

Si bien las categorías anteriores también las podemos encontrar dentro de esta división, puesto que definimos a aquellas guías que están implementadas de forma que sea en un dispositivo externo y ajeno a la institución donde se ofrezca el servicio de la reproducción [5].

En esta línea trabajan diferentes empresas, organismos oficiales y páginas webs que ofrecen archivos de multimedia, no necesariamente sólo audio, descargables por los usuarios para después reproducir en equipos portátiles de los propios turistas [5].

Además encontramos aplicaciones programadas en varios SO para dispositivos móviles, como iPhone o Symbian, mediante el cual gestionar las peticiones de archivos de audio, como puedan ser *GTP Museum Solutions* (iPhone), o *BlueHertz* (Symbian) [5].

■ ■ 2.5.3 Aplicaciones en Dispositivos Móviles

En la actualidad existen diversas fuentes que nos permiten descargarnos información a través de internet, con la finalidad de que nosotros, por nuestra cuenta, reproduzcamos dichos contenidos dónde y cómo creamos adecuado, ya sea en nuestro teléfono móvil, un reproductor de música o cualquier otro medio que acepte archivos de audio.

Debido a la demanda demostrada que este tipo de equipos suscita en los turistas, véase Tabla 2.4, en esta línea trabajan portales como *voices.com*, que nació en el año 2009 y ofrece servicios a una comunidad de usuarios de internet que crean, comparten y escuchan información en formato audio que posee una vinculación geográfica. Además de esta forma de acceder a los archivos de audio, encontramos sistemas que integran esta función dentro de una aplicación para un dispositivo móvil, reduciendo la labor de adquisición de datos previa que requiere lo antes mencionado.

A continuación detallaremos algunas de estas aplicaciones para dispositivos móviles.

2.5. SISTEMAS DE AUDIOGUÍAS

Año	Exposición Permanente			Exposición Temporal		
	Alquileres	Visitas	Uso	Alquileres	Visitas	Uso
2002	32.737	337.371	9,70%	14.755	334.622	4,41%
2003	40.747	380.811	10,70%	9.711	123.985	7,83%
2004	38.844	366.799	8,20%	22.039	316.696	6,96%
2005	36.724	341.155	10,76%	21.517	302.608	7,11%
2006	47.225	407.553	11,59%	27.750	329.201	8,43%
2007	44.303	403.821	10,98%	55.481	527.900	10,51%
2008	40.593	381.619	10,64%	38.715	337.015	11,49%

Tabla 2.4: Visitantes del museo Thyssen de Madrid y uso de Audioguías [7]

2.5.3.1 Alhambra de Granada y The Musseum Of Modern Art

Encontramos algunos ejemplos de aplicaciones similares a la que queremos desarrollar, sobre todo en cuanto a fondo. Por ejemplo, el museo de arte moderno MOMA, o la Alhambra de Granada, son espacios que disponen de una guía para teléfonos *smartphone*.

La Alhambra de Granada pone a disposición de los usuarios una aplicación desarrollada en J2ME, lo que va a requerir al usuario que cuente con un programa capaz de ejecutar este tipo de archivos.

Por su parte, el museo de arte moderno de Nueva York pone a disposición de sus clientes una aplicación de guía de museos sobre el SO iPhone que se asemeja a la que busca desarrollar el presente proyecto.

2.5.3.2 Toozla

Basa su funcionamiento en un programa para móviles que soporte J2ME, o *iPhone*, que nos muestra información puntual acerca de dónde podemos encontrar monumentos famosos, restaurantes, tiendas, museos o teatros. Su uso está orientado a informar al usuario acerca de los

CAPÍTULO 2. ESTADO DEL ARTE

puntos de interés de una ciudad, haciendo uso del posicionamiento GPS si estuviera disponible en el teléfono [48].

■ ■ ■ 2.5.3.3 BlueHertz

Esta compañía propone un sistema llave en mano de guías para móviles, se basa en el envío vía *bluetooth* de todo el programa, incluyendo los archivos con información multimedia, en unos puntos clave, estando disponible para cualquier usuario sin importar horarios. Sufre el inconveniente de tener que enviar toda la información al móvil, con la carga de memoria que ello conlleva. Esta empresa ha desarrollado una solución para el Ayuntamiento de Denia. Siguen una filosofía similar otras empresas como *BLUTYpoint* [49].

■ ■ ■ 2.5.3.4 TourMovil

Aplicación desarrollada en J2ME que dispone de información turística de utilidad para el visitante, incluye mapas, datos históricos, visitas guiadas, alojamientos y restaurantes, aunque la información mostrada es únicamente informativa [50].

■ ■ ■ 2.5.3.5 Buenos Aires, Argentina

La ciudad de Buenos Aires ha elaborado un servicio de guía turística a lo largo de 12 puntos de la ciudad, los cuales pueden descargarse a través de internet, y aquí viene la novedad, también son accesibles por medio de una llamada telefónica y la introducción de un código [51].

■ ■ ■ 2.5.3.6 noTours

Proyecto de la empresa *Escoitar.org*, nos permite recorrer un espacio a la vez que disfrutamos de una experiencia acústica binaural por medio de dispositivos móviles basados en *Android*, haciendo uso también de funcionalidades GPS para posicionar a los usuarios. La particularidad de este sistema radica en la inclusión de sonidos propios del lugar [52].

2.5.3.7 Babia, YeaYe y Android Place Directory

Estos tres productos implementan una guía de viaje con un gran directorio de información para el turista que viaja a una ciudad. Mientras que los dos últimos están desarrollados bajo *Android* y acceden a los contenidos a través de una conexión a internet. El primero utiliza un sistema de llamadas a un número similar al implementado en la ciudad de Buenos Aires [53].

2.5.3.8 wikiTude

Aplicación desarrollada para terminales con SO *Android*, ofrece al usuario una guía turística basada en realidad aumentada, aprovechando la cámara del teléfono, la brújula y el GPS para poder mostrar por pantalla información sobre lo que estamos viendo [54].

2.6 Conclusiones

La ventaja que presentan los *smartphone* frente a otros terminales no tan avanzados es el número de aplicaciones que están continuamente apareciendo en el mercado, además de traer con ellos una complejidad mayor.

Sin embargo, y pese a la continuación aparición de nuevas aplicaciones, entre las guías de museos no existen muchas aplicaciones que realicen una labor similar, a través de un teléfono móvil.



II

Diseño e Implementación



3

Requisitos de la Aplicación

Para la elaboración de este documento hemos seguido las directrices que marca el estándar de la ESA PSS-05-0 Número 2, que data de febrero de 1991, y ha sido revisado en el año 2003 por Vicente Palacios [55], el cual desarrolla una guía para la aplicación de Estándares de Ingeniería de Software para proyectos de software pequeños.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

3.1 Introducción

El documento de la ESA [55] nos propone que para poder aplicar dicho estándar a un proyecto de software éste debe cumplir al menos uno de los siguientes criterios:

- Si se necesitan menos de dos años hombre de esfuerzo para el desarrollo.
- Si se requiere un equipo único de desarrollo de cinco o menos personas.
- Si la cantidad de código fuente es inferior a 10000 líneas, excluyendo los comentarios.

Aunque en este punto no podemos advertir la duración del mismo, ni cuál va a ser el número de líneas de código que se van a necesitar para desarrollarlo, si podemos certificar que para llevarlo a cabo van a intervenir menos de cinco personas, con lo que los requerimientos se cumplen, y por tanto podemos aplicar el documento elaborado por la ESA [55].

Establecida la línea que seguiremos en el desarrollo del software, este estándar nos propone comenzar realizando un pequeño estudio de Casos de Uso del sistema, para continuar enunciando los diferentes Requisitos de Usuario y de Software con que trabajaremos.

3.2 Casos de Uso

Los casos de uso proporcionan los escenarios mediante los cuales definimos como interactuará el sistema con los diferentes usuarios, logrando de esa forma un objetivo específico.

Para comenzar identificaremos los distintos usuarios que van a actuar en este sistema, así pues, podemos encontrar tres usuarios diferentes, que pasarán a definir los actores que intervienen en nuestro software: anónimo, registrado y administrador.

El primero podrá descargarse el programa, darse de alta en la aplicación, ver los comentarios de otros usuarios y acceder a los contenidos; el usuario registrado tendrá la misma experiencia de uso que el anónimo, pero añadiendo además a todo ello la posibilidad de incluir sus comen-

3.2. CASOS DE USO

tarios en los contenidos y puntuar los mismos; por último, el administrador podrá comprobar la funcionalidad del sistema, y además modificar los contenidos que ofrece la aplicación.

Definiremos los casos de uso por medio de Diagramas de Casos de Uso 3.1, y de Tablas de Casos de Uso, que nos definen las características principales que deseamos que el software sea capaz de realizar. A continuación, detallaremos la forma en que estructuraremos la información,

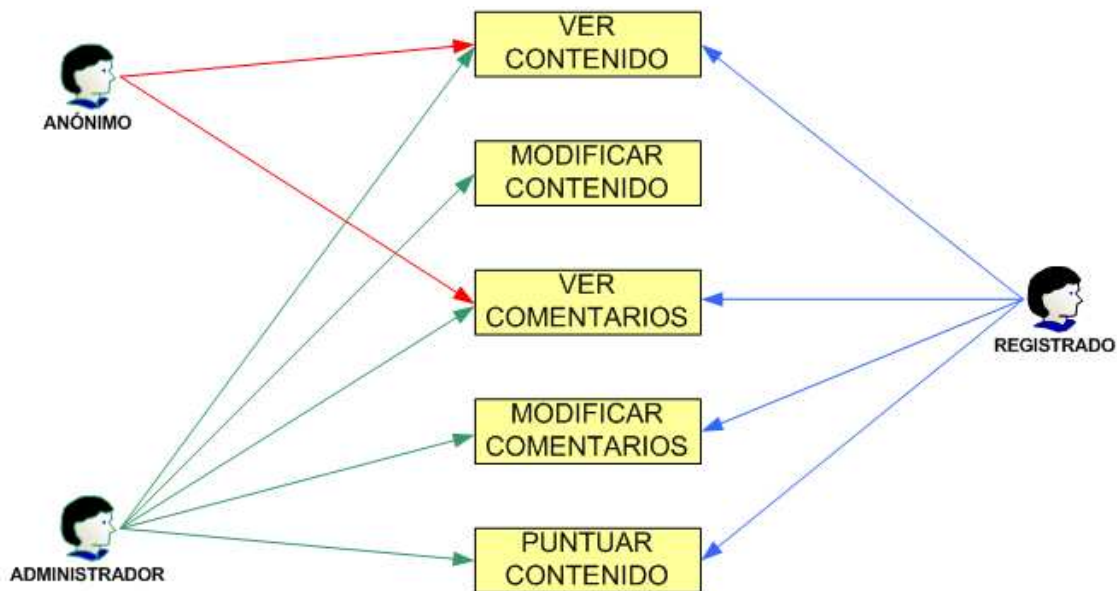


Figura 3.1: Diagrama de Casos de Uso

para continuar comentando cada requisito por separado¹, agrupándolos en los distintos subgrupos que encontramos tanto en los Requisitos de Usuario como en los Requisitos de Software.

¹Para poder comprender mejor los diferentes requisitos, se han realizado tablas para cada uno de ellos, las cuales están disponibles en el Apéndice B y C.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

■ ■ 3.2.1 Diagramas de Casos de Uso

En primer lugar, veremos las diferentes funcionalidades que ofrece el sistema a través de los Diagramas de Casos de Uso, a través de la Figura 3.1 podremos ver de manera más sencilla el funcionamiento global.

■ ■ 3.2.2 Tablas de Casos de Uso

Por medio de las Tablas de Casos de Uso (3.1, 3.2, 3.3, 3.4, 3.5) analizamos de forma más pormenorizada las funcionalidades principales que este sistema ofrece.

CU001	Ver Contenido
Actores	Ánonimo, Registrado, Administrador
Descripción	Reproducimos el contenido multimedia, ya sea audio, vídeo, texto, imágenes o cualquier otro tipo, asociado a la selección que hemos realizado
Precondiciones	Debe existir previamente el contenido, y estar el archivo disponible
Flujo Principal	<ul style="list-style-type: none">■ Selecciono un contenido■ Selecciono el tipo multimedia a reproducir■ Reproduzco/Visualizo el contenido
Postcondiciones	El usuario ha visto el contenido
Excepciones	<ul style="list-style-type: none">■ No existe el fichero que queremos utilizar■ Fallo en la conexión

Tabla 3.1: Caso de Uso CU001, Ver Contenido

3.2. CASOS DE USO

CU002	Modificar Contenido
Actores	Administrador
Descripción	Añadimos, editamos o eliminamos contenido accesible por los usuarios.
Precondiciones	Debe loguearse el administrador
Flujo Principal	<ul style="list-style-type: none">▪ Login de Adminstrador▪ Edición de contenidos
Postcondiciones	Contenidos modificados
Excepciones	

Tabla 3.2: Caso de Uso CU002, Modificar Contenido

CU003	Ver Comentarios
Actores	Ánonimo, Registrado, Administrador
Descripción	Consultamos los comentarios que otros usuarios registrados han dejado sobre un contenido
Precondiciones	
Flujo Principal	<ul style="list-style-type: none">▪ Selección de contenido▪ Ver comentarios
Postcondiciones	El usuario puede observar los comentarios
Excepciones	

Tabla 3.3: Caso de Uso CU003, Ver Comentarios

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

CU004	Modificar Comentarios
Actores	Registrado, Administrador
Descripción	Añadir, modificar o eliminar un comentario por parte de un usuario registrado
Precondiciones	Login como usuario registrado o administrador
Flujo Principal	<ul style="list-style-type: none">▪ Login como registrado o administrador▪ Selección de contenido▪ Selección de comentario▪ Modificación de comentario
Postcondiciones	Comentario modificado
Excepciones	

Tabla 3.4: Caso de Uso CU004, Modificar Comentarios

CU005	Puntuar Contenido
Actores	Registrado, Administrador
Descripción	Puntuamos un contenido o un comentario en función de los criterios del usuario registrado o administrador
Precondiciones	Login como usuario registrado o administrador
Flujo Principal	<ul style="list-style-type: none">▪ Login como registrado o administrador▪ Seleccionar contenido<ol style="list-style-type: none">1. Ver contenido y puntuar contenido2. ver comentario y puntuar comentario
Postcondiciones	Comentario puntuado
Excepciones	

Tabla 3.5: Caso de Uso CU005, Puntuar Contenido

3.3 Identificación de los Requisitos

Los diferentes requisitos con que nos encontramos se dividen en Requisitos de Usuario y Requisitos de Software, que a su vez poseen subcategorías, capacidad y restricción para el primer grupo; y funcionales, interfaz, recursos, documentación y seguridad para el segundo; para las cuales encontramos diferentes requisitos.

Así mismo, en el Apéndice están disponibles todas las tablas que desarrollan cada uno de ellos, en ellas se ha estructurado la información de forma que su comprensión sea más clara y sencilla, así tendremos una plantilla para las tablas que definen estos requisitos que explicamos a continuación:

- **Identificador**

Identifica de forma unívoca cada requisito que definamos, siguiendo la siguiente nomenclatura:

Requisito + Tipo + Plataforma + Número

1. Requisito: Determina el tipo de requisito, Requisito de Usuario (RU) o Requisito de Software (RS).
2. Tipo: De cada tipo de requisito existen varios subtipos, por ejemplo tenemos como Requisitos de Usuario los de tipo de Capacidad y de Restricción.
3. Plataforma: La aplicación que desarrollamos tendrá dos posibles formas de interacción, a través de un navegador web (W), o por medio de un dispositivo móvil (M).
4. Número: Enumera cada requisito por medio de tres dígitos que irán incrementando.

- **Descripción**

Comentario textual de los requisitos.

- **Fuente**

Origen del que se ha obtenido el requisito.

- **Necesidad**

Nivel de necesidad del requisitos dentro del conjunto del sistema final.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

1. Esencial: Requisito indispensable, incluido sin ningún tipo de negociación.
 2. No Esencial: Puede ser objeto de negociación.
- **Prioridad**
Prioridad en el desarrollo del requisito, puede ser Alta, Media o Baja.
 - **Estabilidad**
Indica si el requisito puede cambiar a lo largo de su desarrollo.
 1. Alta: No va a ser modificado.
 2. Baja: Puede variar en las etapas del proyecto.
 - **Verificabilidad**
Muestra la inclusión del requisito en las pruebas de verificación del software.

3.4 Requisitos de Usuario

En este apartado se especifican los requisitos de usuario obtenidos mediante el estudio de la competencia y técnicas de inspección de requisitos (caso de uso descritos anteriormente).

Describiremos el objetivo general de la aplicación, así como el contexto en el que deberá operar la misma, agrupando los requisitos encontrados como Requisitos de Capacidad y Requisitos de Restricción.

■ ■ 3.4.1 Capacidad

Definiremos así los requisitos del sistema traducidos en funciones y operaciones que dotan a los usuarios de herramientas para lograr obtener el rendimiento deseado de la aplicación, de esta forma describe operaciones que el software ha de llevar a cabo.

3.4. REQUISITOS DE USUARIO

■ Parte Web

RUCW001 Accesibilidad a la plataforma web

RUCW002 Login/Logout de Usuario

RUCW003 Autologin de usuario

RUCW004 Descargar información

RUCW005 Editar la información del usuario.

RUCW006 Ver comentarios introducidos por el usuario.

RUCW007 Ver la información asociada a un contenido.

RUCW008 Ver los comentarios asociados a un contenido.

RUCW009 Editar un comentario.

■ Parte Móvil

RUCM001 Instalación de la aplicación.

RUCM002 Iniciar aplicación.

RUCM003 Salir de la aplicación.

RUCM004 Ver Ayuda.

RUCM005 Login/Logout de Usuario.

RUCM006 Autologin de Usuario

RUCM007 Ver contenido.

RUCM008 Puntuar contenido.

RUCM009 Ver comentario.

RUCM010 Modificar comentario.

■ ■ 3.4.2 Restricción

Los requisitos de Usuario de Restricción van a definir restricciones en las que el software es construido y funciona, pero sin alterar los Requisitos de Usuario de Capacidad.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

■ Parte Web

RURW001 Tiempo de respuesta en la carga de nuevas páginas.

RURW002 Tiempo de respuesta en las consultas a la base de datos.

RURW003 Resolución de pantalla.

RURW004 Independencia de SO.

RURW005 Compatibilidad de navegadores.

RURW006 Acceso desde lugares remotos.

RURW007 Parámetros iniciales.

■ Parte Móvil

RURM001 Respuesta en la carga de la aplicación.

RURM002 Respuesta en la carga de nuevos *Activity*.

RURM003 Respuesta en la captura de datos web.

RURM004 Resolución de pantalla.

RURM005 Uso de SO.

RURM006 Compatibilidad de versiones antiguas.

RURM007 Parámetros iniciales.

RURM008 Parámetros configurables.

RURM009 Guardado de información.

3.5 Requisitos de Software

Se elaboran a partir de los Requisitos de Usuario, siendo la base para el diseño y la implementación del sistema final, logrando definir la naturaleza exacta de la aplicación.

De esta forma, nos encontramos diferentes tipos de requisitos: Funcionales, Restricción, Interfaz, Recursos, Documentación y Seguridad, los cuales desarrollaremos a continuación cada uno por separado.

■ ■ 3.5.1 Funcionales

Especifican cuál es la funcionalidad que tiene el sistema, de ese modo definen los propósitos del software en cuestión, siendo estos requerimientos derivados de los Requisitos de Usuario especificados en la sección anterior.

- Parte Web

RSFW001 Consulta de información de contenidos.

RSFW002 Reproducción de contenidos.

RSFW003 Inicio del portal web.

RSFW004 Gestión de credenciales.

- Parte Móvil

RSFM001 Consulta de información de contenidos.

RSFM002 Reproducción de contenidos.

RSFM003 Inicio de la aplicación.

RSFM004 Inicio de la reproducción.

RSFM005 Guardado de credenciales.

RSFM006 Lectura de credenciales.

■ ■ 3.5.2 Rendimiento

Describen límites o condiciones sobre cómo diseñar la aplicación, aunque no deben llegar a suplantarlo dicho proceso, especificando condiciones impuestas por el cliente, el entorno u otros factores sobre el proyecto.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

- Parte Web

RSRW001 Tiempo de respuesta a la selección del usuario.

RSRW002 Tiempo de respuesta a consultas a la base de datos.

RSRW003 Tiempo de respuesta a la carga de contenidos.

RSRW004 Conexión a internet.

- Parte Móvil

RSRM001 Tiempo de respuesta a la carga de *Activity*.

RSRM002 Tiempo de respuesta a la carga de contenidos.

RSRM003 Tiempo de respuesta a consultas a la base de datos.

RSRM004 Conexión a internet.

■ ■ 3.5.3 Interfaz

Desarrolla la forma en que la aplicación se comunica con su entorno, bien en la comunicación con el usuario, o bien en la comunicación con otras aplicaciones.

- Parte Web

RSIW001 Navegadores.

RSIW002 Gestión de la información.

- Parte Móvil

RSIM001 SO a utilizar.

RSIM002 Gestión de la información.

■ ■ 3.5.4 Recursos

Especifican los tipos y la cantidad de recursos que el sistema va a necesitar, desde el uso de memoria hasta el tráfico de datos que soportará.

- Parte Web

RSCW001 Memoria utilizada.

RSCW002 Hardware necesario.

RSCW003 Conexiones necesarias.

- Parte Móvil

RSCM001 Memoria utilizada.

RSCM002 Hardware necesario.

RSCM003 Conexiones necesarias.

■ ■ 3.5.5 Seguridad

Definen la seguridad del sistema frente a amenazas externas o sobre la propia seguridad de las personas, ejemplos de ello son la confidencialidad, integridad o disponibilidad.

- Parte Web

RSSW001 Comprobación de los datos del usuario.

RSSW002 Comprobación en el acceso del usuario.

CAPÍTULO 3. REQUISITOS DE LA APLICACIÓN

- Parte Móvil

RSSM001 Comprobación de los datos del usuario.

RSSM002 Comprobación en el acceso del usuario.

3.6 Conclusiones

Antes de meternos de lleno con el desarrollo de la guía multimedia es una buena práctica estudiar cuales son los requerimientos que tenemos, para de esta forma adaptar lo máximo posible nuestra aplicación a lo que esperaba el usuario encontrarse.

El estándar que define la ESA nos indica unos sencillos pasos a seguir en la adquisición de datos para poder captar toda la información posible sobre el diseño que queremos realizar.



4

Implementación del Sistema

Una vez definidos los Requisitos de Usuario y los Requisitos de Software, el siguiente paso es implementar el Sistema de Guía Multimedia sobre el SO Android, lo que nos llevará a dividir el desarrollo en dos partes, por un lado la creación de la aplicación web, desde la que el administrador modifica los contenidos de la aplicación, y los usuarios accederán a los mismos; y por otro lado la aplicación móvil, que dotará de ubicuidad al usuario del software.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

4.1 Introducción

A lo largo de este capítulo mostraremos los pasos que hemos seguido para la realización del software, para ello explicaremos las decisiones de diseño que se han tomado. Comenzaremos determinando los diferentes lenguajes de programación a utilizar, para después explicar las herramientas usadas en el desarrollo de esta aplicación.

La aplicación que hemos desarrollado a lo largo del presente proyecto se ha dividido en tres partes, cada una de ellas imprescindible para que el sistema funcione correctamente, por ello mostraremos y detallaremos los diferentes diagramas de bloques implementados en la aplicación web, la aplicación sobre Android, y el diagrama que sigue la base de datos.

4.2 Lenguajes de Programación

La elección de los lenguajes de programación varía en función de si hablamos de la aplicación web o de Android, por un lado usaremos el SDK de Android para desarrollar esta parte, puesto que no intervienen elementos gráficos críticos. Por otro lado, programaremos la parte web utilizando PHP y *MySQL*, entre otros, debido a que nos permiten cumplir el objetivo marcado de economizar el desarrollo.

Si bien, aunque Android permite programar en C/C++ haciendo uso del NDK, en nuestro caso los requisitos de procesamiento gráfico no son críticos, y es por ello que usaremos el SDK proporcionado, es decir, para la programación de la parte destinada al teléfono móvil utilizaremos Java. El cual estará complementado por XML que se encargará de definir, entre otros, los diferentes *layout* o el *Manifest* de la aplicación.

Por otro lado, el desarrollo de la parte web de la aplicación requerirá el uso de otros lenguajes de programación que permitan implementar contenido dinámico, así como generar transacciones con una base de datos.

Entre las opciones que tenemos, usaremos *MySQL*, como base de datos, y PHP como len-

4.3. ELECCIÓN DE LAS HERRAMIENTAS

guaje del lado del servidor, lo que nos va a permitir tener contenido dinámico en nuestra web, realizar consultas a dicha base de datos *MySQL*, o manejar sesiones de usuario entre las opciones más destacables.

Si comparamos PHP con otras opciones similares, encontramos que PHP frente a ASP se diferencia en que el segundo sólo puede usarse sobre servidores *Windows*, a diferencia de PHP que puede ser ejecutado sobre *Linux* o sobre *Windows*, lo que supone un abaratamiento del proyecto usar PHP puesto que no nos impone la adquisición de una licencia de software *Windows*.

Siguiendo esta línea comparativa, y partiendo de las diferentes opciones que nos ofrece la empresa HOSPEDAXES [56], podemos ver como la opción más económica para un servidor compartido, con 50 Mb de espacio web y transferencia mensual de 1 Gb, es PHP (26,95 €), frente a ASP (no presenta opción para servidor compartido), JSP (44,95 €) y *Ruby On Rails* (31,95 €).

Añadir, que al estar familiarizado con PHP y MySQL trabajando conjuntamente, el periodo de aprendizaje de las tecnologías necesarias también será menor respecto a los otros lenguajes del lado del servidor y bases de datos *General Public License* (GPL), motivos por los que nos decantamos por el uso de PHP como lenguaje del lado del servidor.

4.3 Elección de las herramientas

Una vez realizada la selección de los diferentes lenguajes de programación que vamos a utilizar para desarrollar esta aplicación, debemos elegir las herramientas que utilizaremos para llevar la misma a cabo.

En primer lugar, usaremos el entorno SDK que Android nos proporciona, que puede ser desarrollado usando el IDE Eclipse, y que además usaremos para programar el resto de lenguajes que intervienen en el proyecto.

Además debemos tener un servidor HTTP que soporte la tecnología del lado del servidor seleccionada, en nuestro caso PHP y *MySQL*. Por ese motivo utilizamos la plataforma XAMP,

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

ya que realizará la instalación y configuración básica para que podamos trabajar con el servidor web *Apache*, junto a PHP y *MySQL*.

A continuación detallaremos con mayor profundidad las características de estas herramientas que nos ayudarán a elaborar la aplicación de Guía Multimedia.

■ ■ 4.3.1 IDE Eclipse

Eclipse fue creado por IBM como una plataforma *Open Source* implementada mediante Java, resaltar que es una aplicación de alta calidad para el desarrollo de herramientas integradas. Actualmente es gestionado por la Fundación Eclipse [57].

Tiene la posibilidad de añadir nuevas funcionalidades al editor, a través de nuevos módulos *plugins* como *Java Development Tools* (JDT), para programar en otros lenguajes de programación además de Java, como C/C++, PHP, *Python* o *Ruby*, además de soportar subversiones [57].

Destacamos sus principales características [57]:

- Multiplataforma (*GNU/Linux*, *Solaris*, *Mac OSX*)
- Soportado para distintas arquitecturas (x86, x86-64)
- Estructura de *plug-in* que hace sencillo añadir nuevas características y funcionalidades.
- Control de versiones con cvs o con subversión (con subclipse).
- Resaltado de sintaxis, autocompletado, tabulador de un bloque de código seleccionado, multitud de utilidades de edición que ayudan enormemente al programador.
- Asistentes para la creación, exportación e importación de proyectos; para generar esqueletos de códigos (templates).

Podemos ampliar información sobre esta herramienta en la web de *Eclipse Foundation* [57], la cual aparece en la bibliografía.

4.3.2 SDK Android

Como ya introdujimos en el Capítulo 2, para poder desarrollar aplicaciones para la plataforma Android debemos tener unos conocimientos del lenguaje Java y XML, ya que serán los que utilicemos para la programación, además necesitaremos disponer de un entorno de programación, que en nuestro caso será el IDE Eclipse, puesto que la *Open Handset Alliance* nos proporciona herramientas para este entorno [4].

El SDK es necesario para desarrollar aplicaciones en Android, que junto a *Android Development Tool* (ADT) nos proporcionan las herramientas necesarias para desarrollar aplicaciones Android junto al IDE Eclipse [58].

Al desarrollar aplicaciones con el IDE Eclipse nos encontramos con dos perspectivas, denominadas vistas [58]:

- Perspectiva Java

Aquí desarrollamos el código Java de la aplicación, ejecutándose en segundo plano JDT que realizará las labores de compilación automática del código.

- Perspectiva *Dalvik Debug Monitor System* (DDMS)

Esta perspectiva realiza las labores de panel de control de un emulador de Android, donde podemos tener el control sobre los procesos que se ejecutan, el sistema de archivos o las conexiones que se han realizado, además nos proporciona un registro de las acciones que realiza dicho emulador.

Aunque Eclipse y ADT nos proporcionan un gran control sobre el desarrollo en Android, hay veces que necesitamos obtener un mayor control sobre nuestro proyecto, y eso lo conseguimos con las diferentes herramientas de que disponemos por línea de comandos [58]:

- *Android Asset Packaging Tool* (AAPT)

Se encarga de procesar el archivo *main.xml* y *R.java*, combina la funcionalidad de *pkzip* y *jar* junto a un compilador de recursos específico.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

▪ *Debug Bridge*

Android Debug Bridge (ADB) nos permite interactuar con el emulador Android directamente desde la línea de comandos a través de una instancia del mismo, funciona como una aplicación cliente servidor basada en TCP gracias a un par de procesos que se ejecutan de fondo tanto en el emulador como en el ordenador del programador.

Para finalizar haremos hincapié en un par de herramientas más que usaremos para el desarrollo en Android, el conjunto de paquetes de Android, y la Máquina Virtual [58].

Los Paquetes de Android son un conjunto de clases necesarias para el desarrollo en Android, algunos de ellos esenciales para la creación de software en esta plataforma, son fácilmente identificables porque el nombre del paquete comienza por Android [58].

Por otro lado, *Android Virtual Machine* (AVM) *Manager* permite crear una máquina virtual Android en la cual poder probar las aplicaciones que desarrollamos, proporcionando distintos *skins* con tamaños y orientaciones variados, definiéndole además diferentes características como memoria, tamaño de disco o velocidad de la red [58].

■ ■ 4.3.3 XAMP

Esta distribución tiene por objeto facilitar la labor de instalación de varias herramientas necesarias en el desarrollo web, entre las que se encuentran *Apache*, *PHP* o *MySQL*, configurando todas ellas para su correcto funcionamiento de forma conjunta, estando disponible en diferentes distribuciones, *Linux*, *Windows*, *Mac OS* y *Solaris*. Además, Distribución que instala y configura Apache, MySQL y PHP en Windows (XAMP) se distribuye bajo licencia GNU GPL [59].

XAMP nos ayuda a la hora de configurar varios elementos que utilizamos para el desarrollo del presente proyecto, como son Apache y MySQL y PHP, los cuales comentaremos con algo más de detenimiento [59].

4.3. ELECCIÓN DE LAS HERRAMIENTAS

4.3.3.1 Apache

Dentro del proyecto *Apache HTTP Server*, de la *Apache Software Foundation*, se desarrolla el servidor *Apache*, que supone un esfuerzo por mantener un servidor HTTP de código abierto para diferentes SO que permite implementar HTTP 1.1 [9].

Apache es modular, por lo que nos ofrece muchas posibilidades de configuración; de código abierto, con lo que su distribución es gratuita; también es multiplataforma, con lo que puede usarse en diversas plataformas como *Linux* o *Windows*; además es extensible, con diversos módulos accesibles por los usuarios del servidor; y altamente utilizado en todo el mundo [9].

Aunque su instalación puede llegar a resultar tediosa, XAMP la simplifica para que podamos trabajar directamente con el mismo una vez se haya instalado la distribución, aunque podemos acceder a los archivos de configuración del servidor para modificar todas las directivas que en él se encuentran [9].

Según datos de *NetCraft* [60] de Agosto de 2010, *Apache HTTP Server* sigue siendo el servidor web más utilizado en el mundo gracias a su robustez, extensibilidad y rapidez, ocupando algo más del 50 % [9].

4.3.3.2 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario, rápida, segura, fácil de usar, y con un gran número de usuarios que la utilizan, donde los derechos del código están en posesión de *MySQL AB*, quién desarrolla *MySQL* como software libre, ofreciendo *MySQL* bajo licencia GNU GPL, aunque su uso en software privado requiere la adquisición de una licencia [9].

MySQL usa el lenguaje SQL estandarizado para el almacenamiento, actualización y acceso a información. *MySQL* es muy rápido y capaz de almacenar grandes cantidades de datos. Además, soporta muy diversos lenguajes de programación diferentes, *C*, *C++*, *Eiffel*, *Java*, *Perl*, *PHP*, *Python* y *TCL* [9].

4.4 Aplicación Web

La parte web de la aplicación está desarrollada en dos secciones, por un lado tenemos los contenidos sin ningún tipo de restricción de uso o acceso, y por otro los que necesitan que el usuario se haya dado de alta en la aplicación.

La primera parte muestra contenidos completamente estáticos por los que el usuario puede navegar y conocer con mayor profundidad en qué consiste la herramienta desarrollada, permitiendo, además, acceder al alta de usuario, formulario de acceso o recordatorio de contraseñas entre otros.

Además, destacar que no todos los usuarios tendrán acceso a las mismas zonas dentro de la parte privada, los usuarios podrán consultar los contenidos así como sus comentarios realizados, pero sólo el administrador podrá añadir, modificar o eliminar cualquier tipo de contenido a través de dos enlaces únicamente visibles para él.

■ ■ 4.4.1 Zona Pública

Mostraremos el esquema de las diferentes partes que posee la web, tal y como ilustra la Figura 4.1, diferenciando mediante el borde de cada casilla si el acceso es público o privado, y en caso de ser privado que tipos de usuarios podrán acceder a ellos.



Figura 4.1: Esquema del portal web 9GM

El color azul denota aquellas *Uniform Resource Locator* (URL) que son accesibles para todos los usuarios que visiten la web, siendo el resto de páginas de carácter privado, a excepción de las páginas con el texto en otros idiomas que están sin implementar, y aparecen en color negro.

Por su parte el color rojo nos informa de que sólo el administrador tendrá la capacidad de entrar en dichas páginas, el color naranja está reservado para URLs a las que pueden acceder cualquier usuario registrado (administrador o no), y las de color amarillo son para usuarios registrados a excepción del administrador.

Para la navegación por los contenidos se han habilitado dos menús diferentes: el primero dedicado a los contenidos públicos, que está disponible en la parte superior e inferior de la página, así como en el lateral derecho del área de navegación; el segundo únicamente es accesible cuando el usuario ha iniciado sesión, localizándose en la parte derecha de la pantalla debajo del menú general. A continuación describimos la función que cada página realiza:

- Inicio

La página de inicio posee la labor de brindar la bienvenida a los usuarios que se dirigen hasta ella, en la misma encontramos información sobre el sitio: qué se está visitando, así como acceso directos y visuales a las principales partes del sistema.

- Qué es 9GM

Aquí describimos qué es la aplicación, cómo funciona, para qué puede utilizarse, de dónde proviene la idea y por qué implementarla en un teléfono inteligente.

- Manuales

El usuario encontrará en esta página información de ayuda para poder sacar el máximo rendimiento al sistema, por medio de documentos descargables en forma *Portable Document Format* (PDF).

- Descargas

Si un usuario quiere comenzar a utilizar la aplicación, únicamente debe visitar esta página, ya que aquí podrá descargar la misma para las diferentes plataformas existentes para teléfonos móviles.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

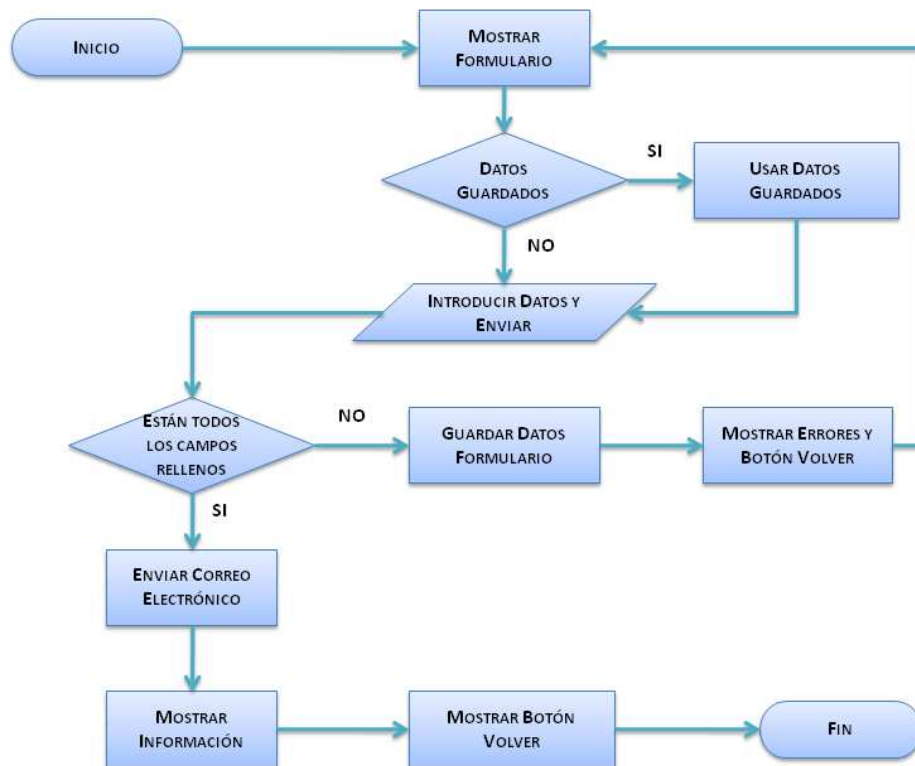


Figura 4.2: Diagrama de flujo del formulario de contacto

■ Imágenes

Las capturas de pantalla ayudan al usuario a hacerse una idea más veraz de la apariencia de la aplicación, así como de sus funcionalidades.

■ Contacto

Aquí encontrará un formulario para que el usuario pueda contactar con el equipo de la aplicación, siendo parte fundamental del proceso de calidad de toda organización mantener un contacto continuo con los consejos, quejas y sugerencias que de su actividad se desprenden.

En la Figura 4.2 encontrará el diagrama de flujo que sigue esta funcionalidad. En primer lugar es mostrado al usuario un formulario al pulsar en la pestaña Contacto, el cual deberá

rellenar por completo para posteriormente pulsar sobre el botón enviar, si no se rellenasen todos los campos, la aplicación lo detectará e informará al usuario sobre el error, mostrando un botón para volver al formulario anterior, donde los campos que se habían introducido correctamente guardan dichos valores.

Cuando se hayan introducido todos los campos correctamente y el usuario haya pulsado el botón enviar, el sistema enviará un correo electrónico al administrador con la información introducida por el usuario, y confirmará a este último que todo ha funcionado correctamente y su mensaje ha sido enviado.

- Zona Privada

Únicamente estará disponible esta sección, y las que dependen de ella, una vez haya iniciado sesión el usuario, pudiendo ver unos contenidos u otros en función de si es administrador o no.

- *English*

La zona de idiomas se encuentra en proceso de construcción, albergará los mismos contenidos que la parte en español, pero traducidos y editados al inglés, ampliándose posteriormente el número de idiomas diferentes disponibles.

■ ■ 4.4.2 Zona Privada

Podemos dividir los contenidos en esta parte del portal web en los siguientes:

- Acceso de Usuarios

Como adelantamos antes, para acceder a la zona privada es imprescindible que el usuario se encuentre registrado en la base de datos de la aplicación, y para ello se le ofrece un mecanismo para darse de alta, así como para recordar la contraseña en caso necesario.

En la Figura 4.3 encontramos el diagrama de flujo que se sigue cuando alguien decide darse de alta en la aplicación, así como si quiere modificar los datos almacenados en la misma por medio de la opción del menú privado Datos Personales.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

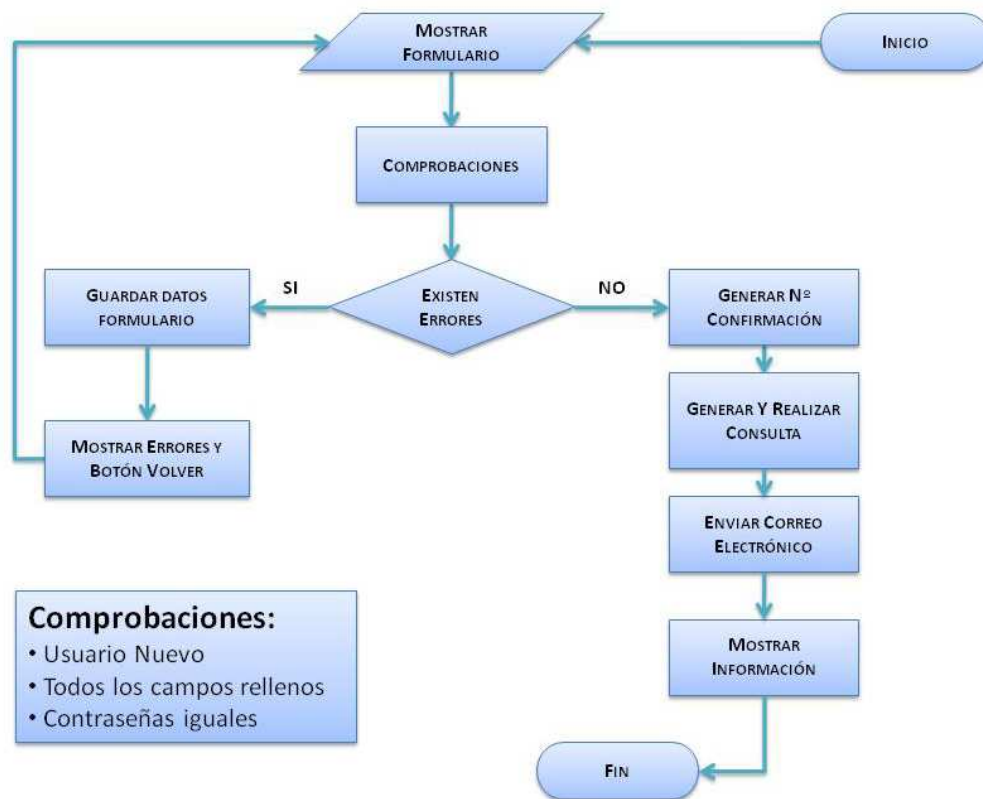


Figura 4.3: Diagrama de flujo para añadir o modificar usuarios

El primer paso será comprobar si el usuario existía en la base de datos, ya que en ese caso se importarán los datos guardados en la misma.

A continuación se mostrará un formulario que el usuario tendrá que rellenar (si ya se había dado de alta aparecerán los datos que se tenían almacenados), finalizando al pulsar el botón enviar. Si todos los campos están correctamente rellenos, y las dos contraseñas coinciden (la segunda es usada para confirmar que no se ha equivocado al introducirla la primera), se procederá al envío de un correo electrónico para que confirme los datos que ha introducido en la base de datos.

Para realizar el proceso de confirmación de datos se generará un número aleatorio que será guardado junto al resto de datos por medio de una consulta a la base de datos, si este paso es correcto se enviará un correo electrónico a la cuenta facilitada con un enlace para confirmar su cuenta.

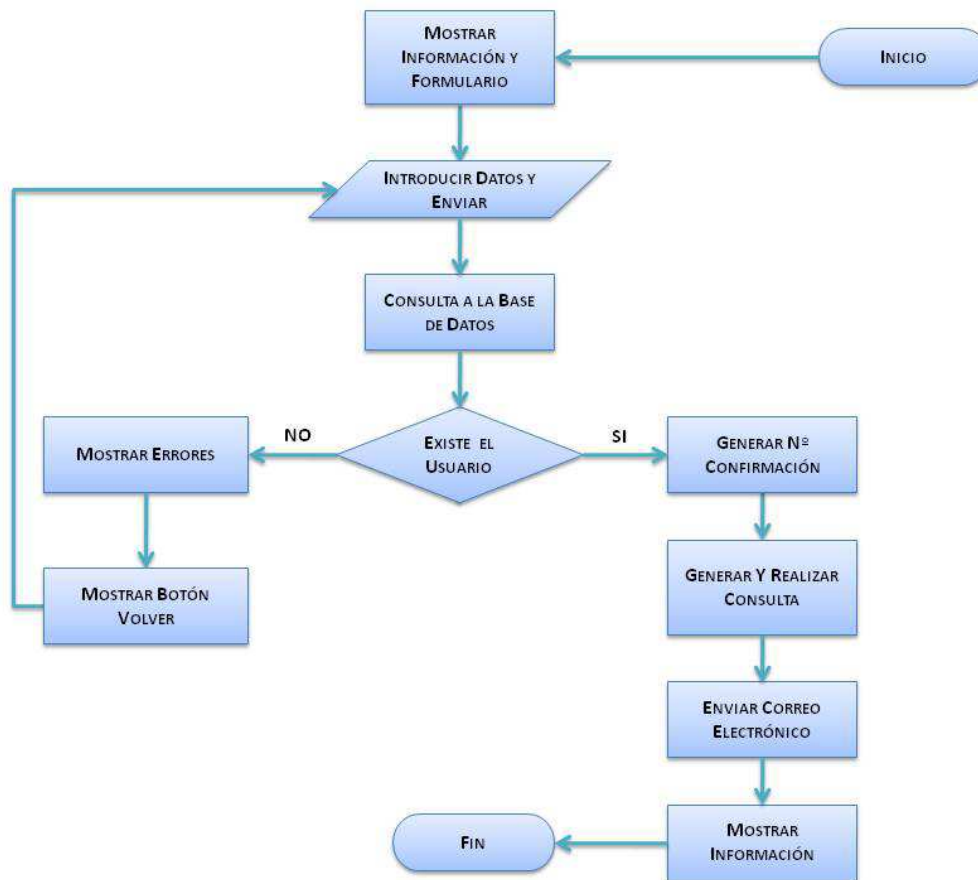


Figura 4.4: Diagrama de flujo para recordar contraseña

En todo momento el usuario conocerá el estado de todo el proceso por medio de la información que se le muestra en el navegador, además, hasta que no confirme su cuenta no podrá entrar en la zona privada, evitando de esta forma que los usuarios se registren con correos electrónicos ajenos a ellos.

Por otro lado, también puede ocurrir que el usuario haya olvidado su nombre de usuario y contraseña para acceder a la zona privada, momento en el que pulsará el enlace que aparece en el menú privado para recordar la contraseña.

El proceso que sigue el recordatorio de contraseña se muestra en la Figura 4.4, comenzamos mostrando un formulario con un campo donde introducir el nombre de usuario y un botón para enviar la información y confirmación sus datos. Para ello se realiza una consulta a

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

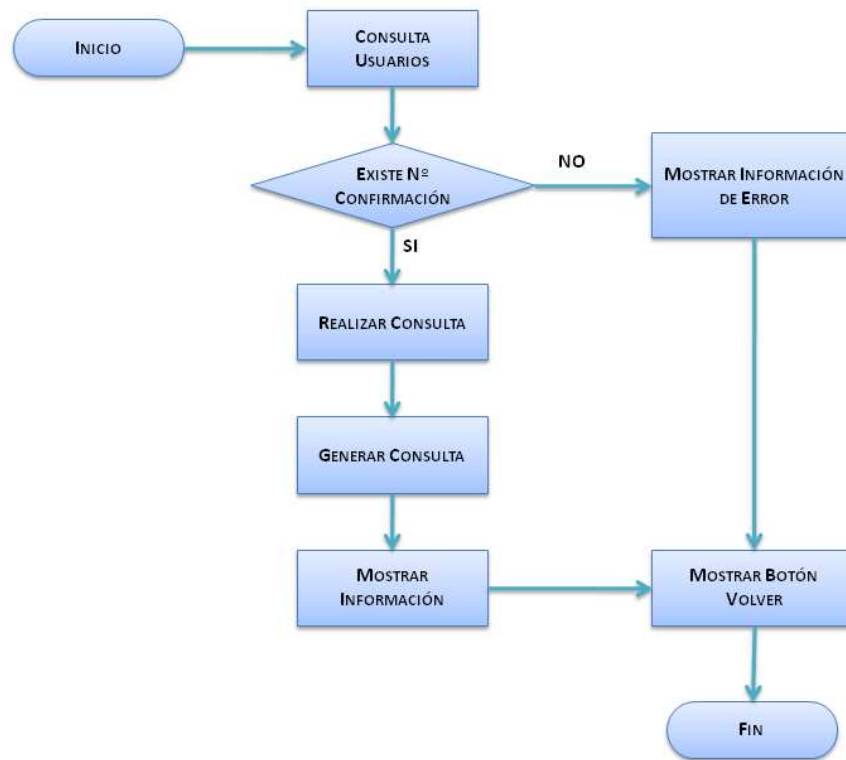


Figura 4.5: Diagrama de flujo para la confirmación de cuenta

la base de datos para saber si ese nombre de usuario existe, mostrando los errores que ha cometido el usuario, o enviando la información necesario al correo almacenado en caso de que todo haya salido de correctamente.

Si se precisa recordar la contraseña, al igual que ocurría cuando se realiza el alta de usuario, es necesario confirmar la cuenta de usuario para poder acceder a los contenidos privados.

Por su parte, cuando realicemos la confirmación de la cuenta introduciendo la URL que nos indica el correo electrónico que hemos recibido se realizarán los pasos descrito en la Figura 4.5.

Habrà una consulta a la base de datos para conocer si el número de confirmación coincide con el almacenado, si todo es correcto el usuario aparecerà como activo en la base de datos, pudiendo acceder a los contenidos de la zona privada; en caso negativo se mostrarà información con el error ocurrido.

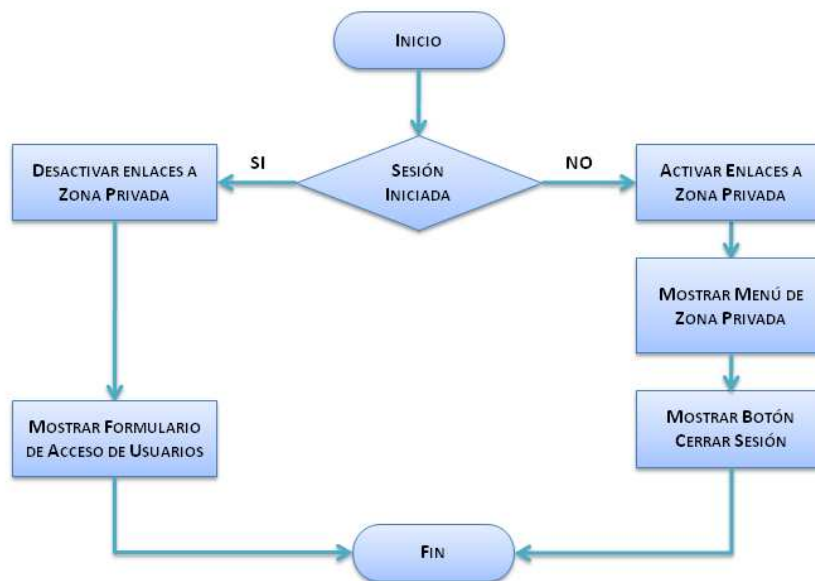


Figura 4.6: Diagrama de flujo para menú de zona privada

■ Menú Privado

El menú privado se muestra en todas las páginas del portal web de 9GM, aunque varía en función de si el usuario ha iniciado sesión.

Tal y como muestra la Figura 4.6, si no se ha iniciado la sesión se mostrará un formulario para realizar la misma, y en caso de haberla iniciado con anterioridad serán mostradas las diferentes secciones disponibles para el usuario, siendo diferentes en caso de tratarse del administrador del portal.

Justamente es en el menú de la zona privada donde cada usuario puede acceder a su cuenta, aunque la comprobación de si se ha iniciado la sesión es anterior, y también se realiza en todas las páginas.

Al cargar cualquier página se inicia la comprobación de inicio de sesión, cuyo diagrama de flujo podemos apreciar en la Figura 4.7. Tenemos la opción de iniciar, cerrar o no modificar la sesión actual.

El primer caso, iniciar sesión, realizará una consulta a la base de datos para comprobar que el nombre de usuario y la contraseña son correctos, iniciando la sesión si no se ha producido error. Si se ha producido algún fallo, se mostrará información sobre el mismo,

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

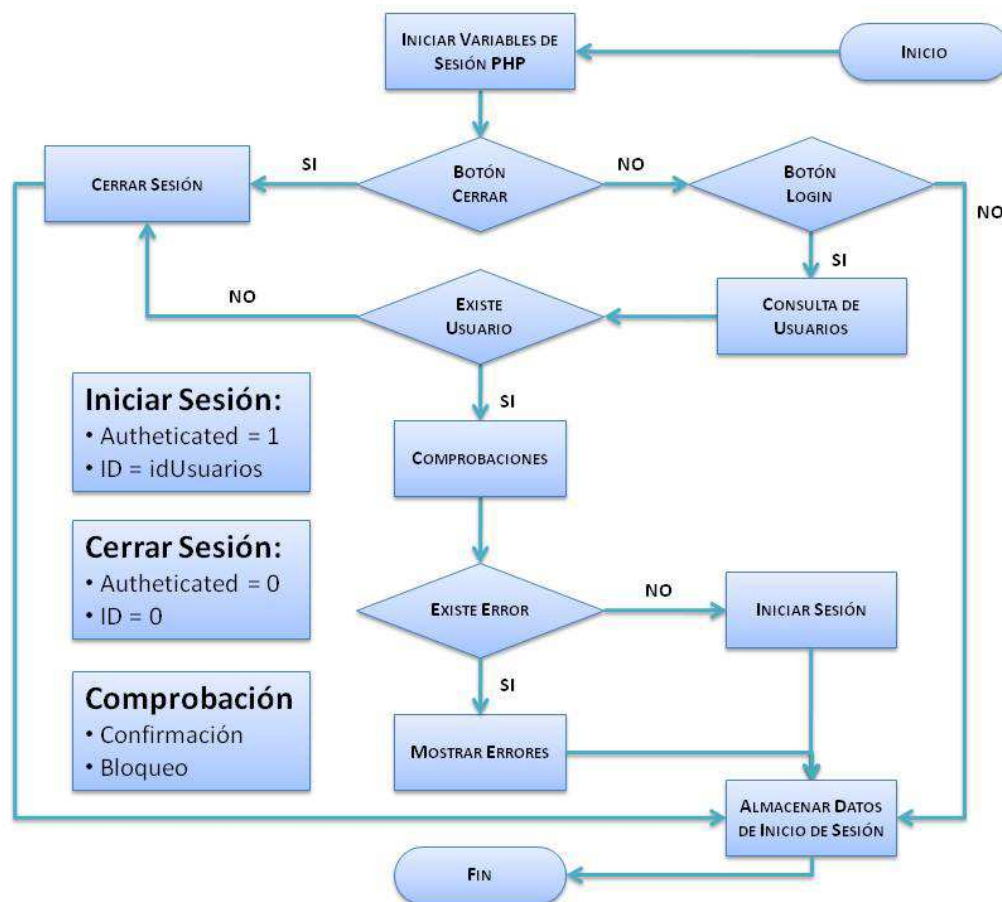


Figura 4.7: Diagrama de flujo para comprobar la sesión

y además se mantendrá la sesión como no iniciada. En el caso que se desee cerrar la sesión se modificarán las variables de sesión que proporciona PHP para que se reconozca como cerrada.

■ Datos Personales

Nos muestra nuestros datos personales almacenados en la base de datos de la aplicación, permitiendo, además, modificar los mismos por medio de un formulario, como hemos descrito anteriormente.

■ Contenidos

Esta página puede considerarse una de las más importantes para el usuario, puesto que a través de ella puede conocer los diferentes contenidos disponibles, así como visualizarlos o

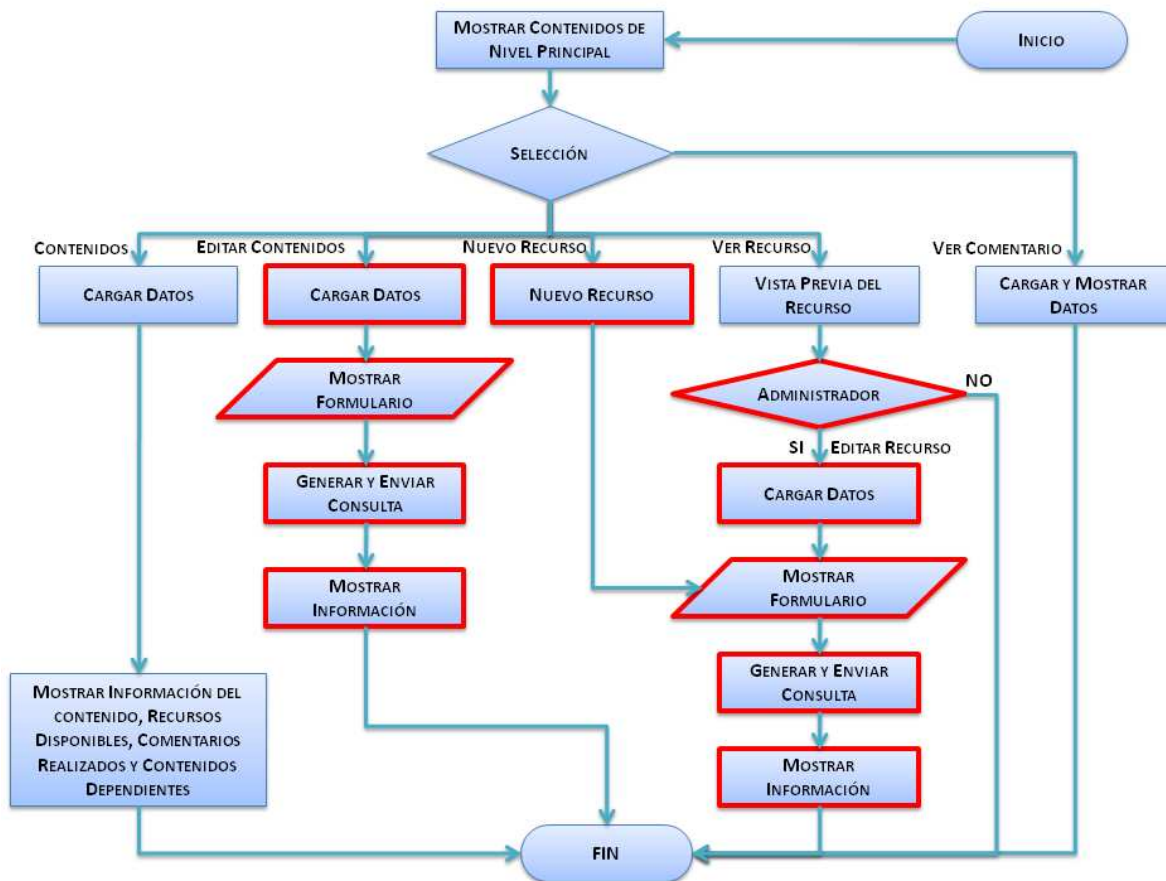


Figura 4.8: Diagrama de flujo para la opción contenidos

consultar los comentarios acerca de ellos.

Si nos fijamos en la Figura 4.8 veremos el diagrama de flujo que sigue esta página. Cuando es cargada la página se muestran todos los contenidos de nivel superior, al pulsar sobre uno de ellos se desplegarán diferentes opciones que podemos seleccionar.

Aparecerá el nombre del contenido, seguido de una breve descripción del mismo; debajo estarán los recursos multimedia disponibles para ese contenido; los comentarios realizados acerca de ese contenido, junto a la puntuación dada, la fecha en que se realizó y el recurso visto por quién escribió el comentario; y la lista de contenidos.

Si seleccionamos un contenido los datos mostrados cambiarán, de forma que podamos ver la información, los recursos multimedia y los comentarios sobre ese contenido. Además, la lista

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

de contenidos se actualizará, añadiendo esta vez aquellos que dependen del seleccionado.

Por otro lado, si pulsamos sobre un recurso, este se mostrará al principio de la página, seguido de toda la información referente al contenido.

La última opción que un usuario tiene es seleccionar un comentario, cargándose la página de nuevo, y mostrando la misma información que antes a excepción de los comentarios, que estará desplegado aquel que se ha seleccionado, pudiendo leerse la opinión suscitada por quién lo escribió.

Además de lo anterior, el administrador tendrá varias opciones más que le permitirán editar los contenidos y los recursos guardados.

En primer lugar podrá añadir contenidos por medio de un formulario cuyo enlace únicamente está visible si ha iniciado sesión como administrador, enviándose la información a la base de datos para que los cambios surtan efecto. Así mismo podrá modificar un contenido, por medio de un formulario similar al anterior, donde se ha añadido la posibilidad de cambiar el contenido superior del que depende.

También podrá añadir o editar recursos por medio de otro formulario, que le permite seleccionar el contenido al que pertenece, así como la dificultad, el tipo de recurso (Texto, Imagen, Audio, Vídeo o Audioguía), una pequeña descripción del mismo y el archivo que contiene el recurso. Una vez pulsado el botón Enviar, se almacenará la información en la base de datos, y el archivo será subido por HTTP al servidor.

■ Visto, Comentado y Puntuado

Por medio de este enlace podremos tener presente los diferentes contenidos que hemos vistos, así como los comentarios dejados al respecto y la puntuación que nos suscitó dicho contenido.

Si miramos la Figura 4.9 veremos que al iniciar esta página se nos muestra una lista con los contenidos que hemos visto y posteriormente hemos comentado y puntuado, pudiendo ver más información a través del enlace.

Si pinchamos en dicho enlace, se mostrará toda la información referente a ese comentario en un formulario, dado que previamente hemos cargando los datos almacenados en la base de datos; pudiendo el usuario modificar el comentario.

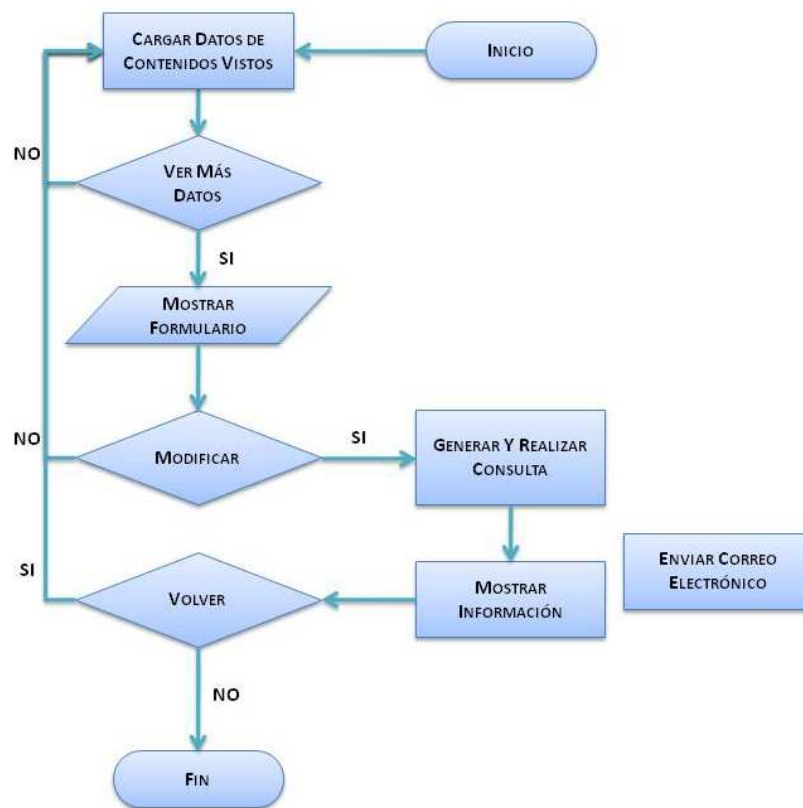


Figura 4.9: Diagrama de flujo para los contenidos vistos, comentados y puntuados

■ Baja de Usuario

Sólo está disponible para los usuarios, puesto que la figura del administrador del sistema ha de estar siempre vigente. Como su nombre indica nos permitirá dar de baja la cuenta del usuario que haya iniciado sesión.

Como vemos en la Figura 4.10, una vez hemos seleccionado darnos de baja, y después de confirmar nuestra decisión, se realiza el borrado de todas las entradas que ese usuario tiene en la base de datos: Votaciones, Contenidos Vistos, Comentarios y su Cuenta de Usuario.

■ Lista de Usuarios

Enlace únicamente accesible al administrador, por medio del mismo podremos ver los usuarios que hay dados de alta, bloquear a un usuario o editar cualquier comentario que hayan realizado.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

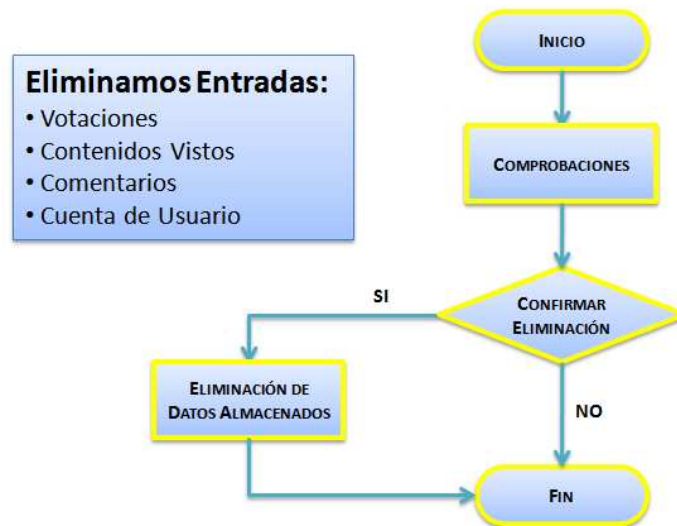


Figura 4.10: Diagrama de flujo Eliminar Usuario

Su modo de funcionamiento se describe en la Figura 4.11, en primer lugar recoge de la base de datos la lista de usuarios registrados que existen, y nos muestra tres opciones en cada uno de ellos: Contenidos Vistos, Bloquear y Contactar.

Si elegimos Contenidos Vistos nos mostrará una lista de los contenidos que ese usuario ha realizado un comentario, posibilitando modificar el comentario realizado por el usuario por medio de un formulario que almacenará la información en la base de datos, y notificará al usuario el cambio por medio de un correo electrónico. Destacar que la puntuación no se puede variar puesto que alteraríamos directamente la opinión de un usuario, no así en un comentario donde puede ser necesario modificar una palabra por calificarla de malsonante.

Si seleccionamos Bloquear, el usuario no podrá acceder a la zona privada ni realizar ninguna acción que requiera estar registrado, anunciándole que se ponga en contacto con el administrador del sitio.

Por último, podemos enviar un correo electrónico a un usuario seleccionando la opción de Contactar, apareciendo un formulario donde introduciremos el texto que será enviado al usuario, recogiendo el resto de información necesaria para el envío de la base de datos.

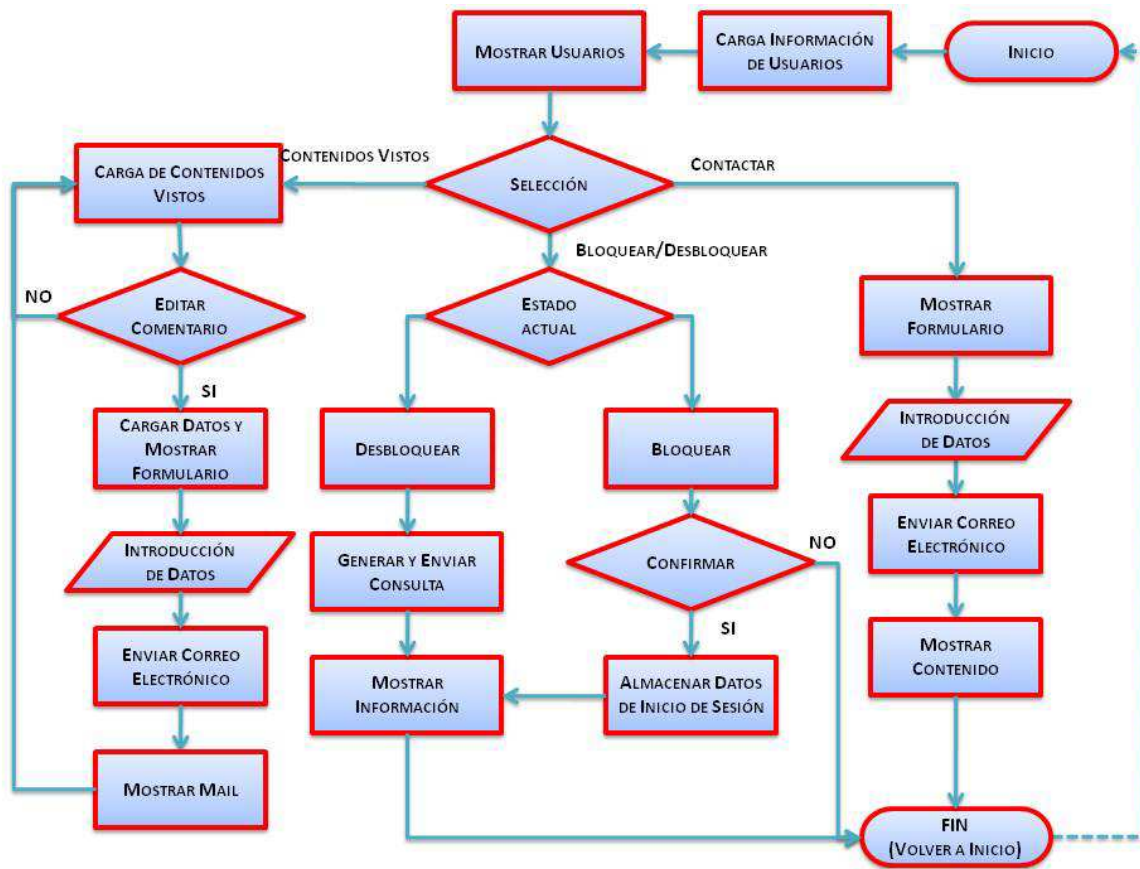


Figura 4.11: Diagrama de flujo para la lista de usuarios

■ Base de datos

Esta página es sólo accesible para el administrador, aquí encontraremos toda la información contenida en la base de datos de la aplicación, contenidos, usuarios, comentarios o votaciones, pudiendo editar las mismas.

En esta sección diferenciaremos dos partes, por un lado los comentarios, que podrán ser modificados o eliminados, y por otro usuarios, contenidos, recursos, contenidos vistos y votaciones, que únicamente se podrán eliminar.

El funcionamiento está descrito en la Figura 4.12, los elementos que sólo se pueden eliminar son cargados de la base de datos, y mostrados mediante una lista, al pulsar sobre el enlace Eliminar se pedirá la confirmación, eliminándose si ésta es aceptada.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

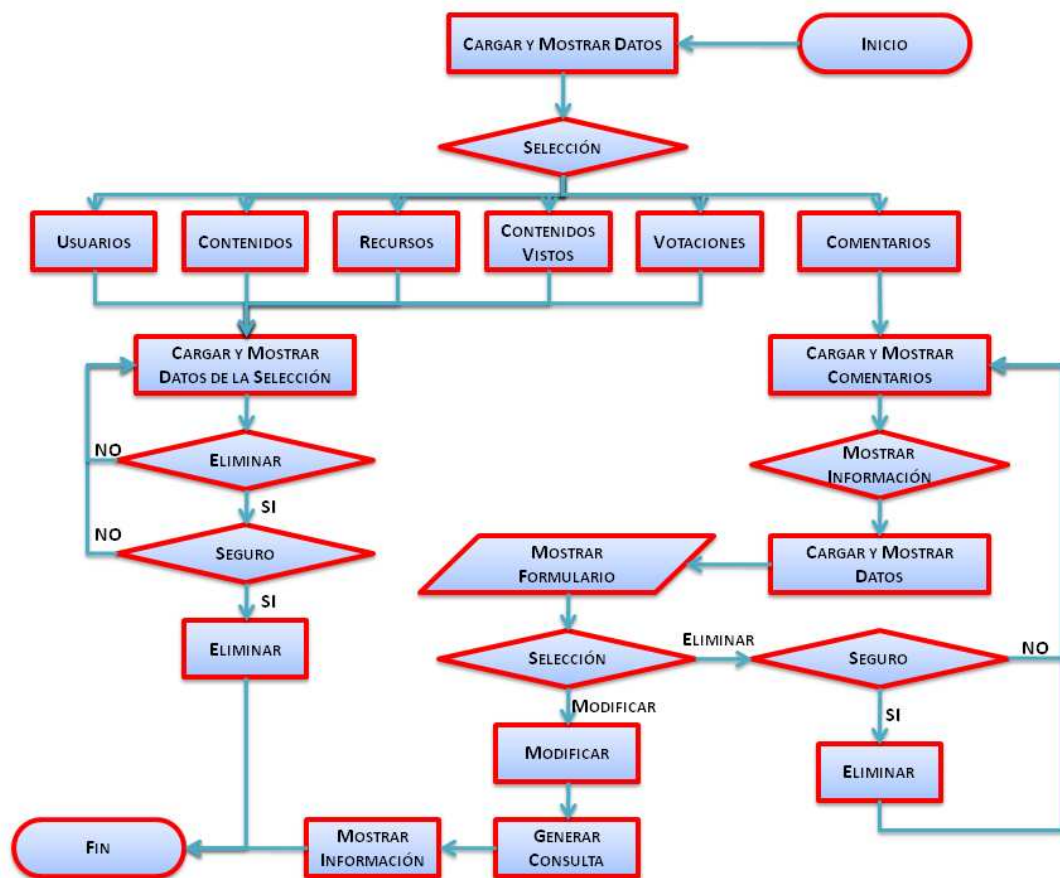


Figura 4.12: Diagrama de flujo para la opción base de datos

Sin embargo, la lista de comentarios nos permitirá mostrar más información, que una vez pulsado sobre el enlace nos mostrará un formulario, con el texto del comentario cargado, y las opciones de modificar el texto o eliminar el comentario, con su correspondiente confirmación.

4.5 Base de Datos

Otra parte muy importantes es el almacenamiento de la información en un servidor, para futuras consultas o modificaciones, labor realizada por la base de datos implementada.

Para la realización de este proyecto se ha optado por utilizar una base de datos *mySQL*, dicha base de datos nos permite realizar consultas por medio del lenguaje *Structure Query Language* (SQL), teniendo, además, completo soporte en el lenguaje que utilizamos en el lado del servidor, PHP.

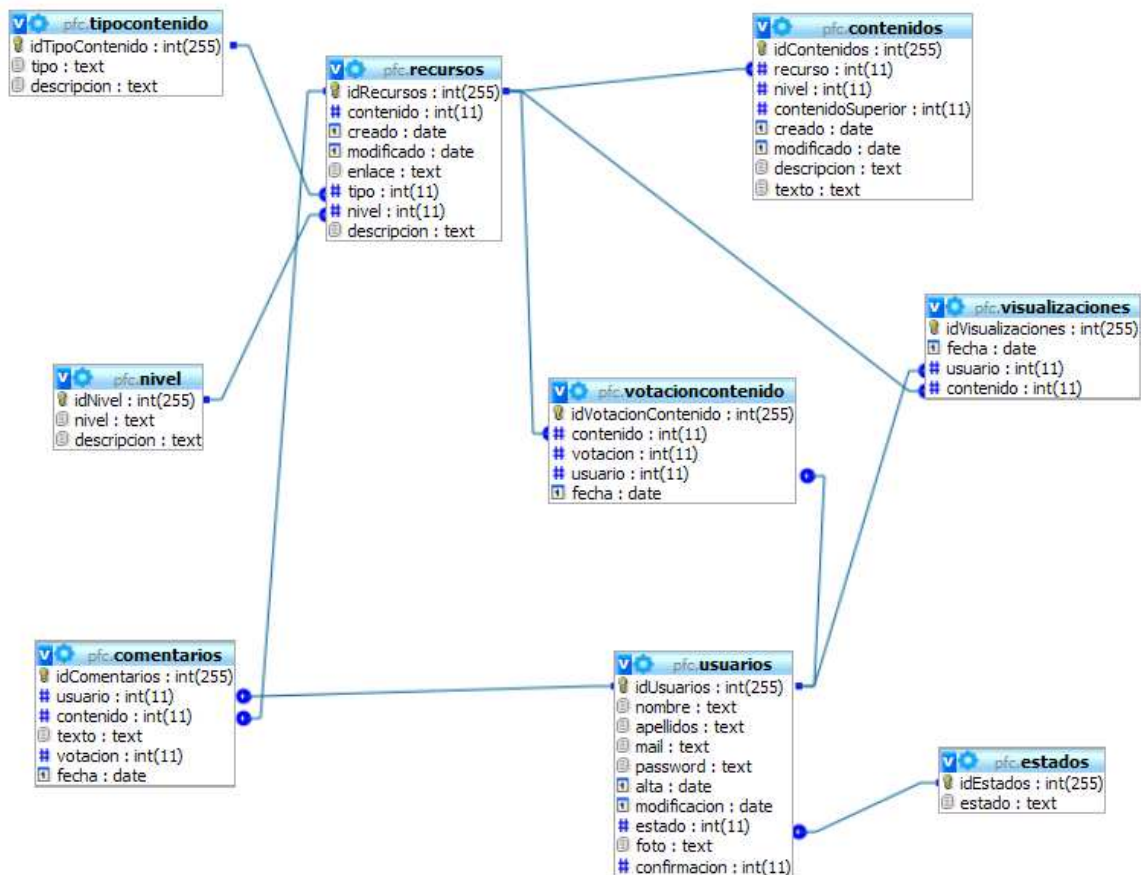


Figura 4.13: Esquema de la Base de Datos de la Aplicación

En el proyecto actual, existen 9 tablas diferentes, como puede apreciarse en la Figura 4.13, con relaciones entre ellas. Las tablas son de tipo MyISAM¹, y el cotejamiento de los campos de texto de las tablas se realiza mediante UTF8. Además, todas las tablas cuentan con una clave principal con autoincremento que hará que las relaciones entre tablas sean unívocas.

Comenzaremos hablando de la tabla Usuarios, que podemos ver en la Tabla 4.1, la clave

¹MyISAM, es la tecnología de almacenamiento usada por defecto en *MySQL*, cuya principal característica es la velocidad de consulta.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

Campo	Clave Principal	Tipo
idUsuario	Si	Entero
nombre	No	Texto
apellidos	No	Texto
mail	No	Texto
password	No	Texto
alta	No	Fecha
modificacion	No	Fecha
estado	No	Entero
foto	No	Texto

Tabla 4.1: Tabla Usuarios

principal es idUsuarios, posee 5 elementos de tipo texto, dos de tipo fecha y dos elementos de tipo entero, el mencionado índice y estado que indica la clave de la tabla Estados a la que hace referencia.

Su labor es almacenar los valores necesarios para cada usuario, siendo sus relaciones con otras tablas las que siguen a continuación:

- Tabla Estados

Para conocer si un usuario está activo, bloqueado o es administrador se necesita conocer el valor de la entrada de la tabla Estados que indica el índice almacenado en la Tabla Usuarios. La relación es de uno a uno, puesto que un usuario únicamente tendrá un estado.

- Tabla Visualizaciones

Cuando un usuario ve un recurso multimedia se genera una entrada en la tabla Visualizaciones, donde el campo usuario de ésta tabla albergará el índice idUsuarios correspondiente a quién observó dicho contenido. Como un mismo usuario puede ver varios contenidos la relación será de uno a varios.

Campo	Clave Principal	Tipo
idComentarios	Si	Entero
usuario	No	Entero
contenido	No	Entero
texto	No	Texto
votacion	No	Entero
fecha	No	Fecha

Tabla 4.2: Tabla Comentarios

- Tabla Comentarios

Al igual que ocurría con la relación anterior, en la tabla Comentarios se guarda el valor del índice de la persona que lo realizó. La relación es de uno a varios.

- Tabla VotacionContenido

Esta relación sigue los pasos de las dos anteriores, guardando cada entrada el valor de la persona que realizó la votación a través de su idUsuario. La relación por tanto será de uno a varios.

La siguiente tabla sobre la que hablaremos será la tabla Comentarios, cuyos campos se definen en la Tabla 4.2. La clave principal es idComentarios, posee un elemento de tipo texto, otro de tipo fecha y cuatro de tipo entero, siendo los campos usuario, contenido y votación el valor de la clave principal de cada tabla a la que hacen referencia.

Campo	Clave Principal	Tipo
idEstados	Si	Entero
estado	No	Texto

Tabla 4.3: Tabla Estados

Su característica principal es almacenar los datos correspondientes a los diferentes comentarios, así como a que usuario pertenece cada uno de ellos y sobre que contenido fue realizado.

Posee dos relaciones, una con la tabla Usuarios, que ya ha sido comentada, y otra con

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

la tabla Recursos, donde cada entrada indica la clave principal del recurso sobre el que se ha comentado. La relación por tanto será de varios a uno, ya que un mismo recurso puede tener varios comentarios distintos.

Por su parte, la tabla Estados se compone de una clave principal de tipo entero y de un campo texto (ver Tabla 4.3). En ella se almacenan los diferentes estados que puede tener un usuario. Su única relación es con la tabla Usuarios que hemos comentado con anterioridad.

Otra tabla importante es la tabla Recursos (ver Tabla 4.4), donde almacenamos las diferentes opciones de visualización que posee cada contenido, así pues diremos que un contenido puede tener varios recursos (audio, vídeo, texto, imagen, audioguía) de dificultades distintas.

Campo	Clave Principal	Tipo
idRecursos	Si	Entero
contenido	No	Texto
creado	No	Fecha
modificado	No	Fecha
enlace	No	Texto
tipo	No	Entero
nivel	No	Entero
descripción	No	Texto

Tabla 4.4: Tabla Recursos

Esta tabla se compone de tres campos de texto, otros dos de fecha y tres de tipo entero, donde tipo y nivel indican el índice de dichas tablas con que se relaciona la tabla Recursos, las cuales podemos ver a continuación:

- Tabla Contenidos

Como adelantamos antes, un contenido puede tener varios recursos, con lo que tendremos una relación de uno a varios.

- Tabla Visualizaciones

Cuando un recurso es visto por un usuario se registra en la tabla Visualizaciones, pudiendo

verse un mismo recurso varias veces, y por usuarios diferentes. Su relación será de uno a varios.

- Tabla VotacionContenido

Si un usuario desea puntuar un contenido después de ver un recurso del mismo, se almacenará en la tabla VotacionContenido el índice del recurso sobre el que se ha votado, siendo su relación de uno a varios.

- Tabla TipoContenido

Puesto que un recurso puede ser de diversos tipos, se hace necesario declararlo en la base de datos, almacenándose en Recursos el valor de la clave principal del tipo de contenido que es, siendo la relación uno a varios.

- Tabla Nivel

Al igual que ocurre con los tipos de contenidos, también existe la opción de que un recurso tenga diferentes niveles de profundidad en la materia que está tratando, existiendo una relación de uno a varios entre las tablas.

- Tabla Comentarios

Aunque esta relación la hemos explicado anteriormente, resaltaremos de nuevo que es necesario saber sobre que recurso, y en consecuencia que contenido, ha sido realizado el comentario.

Campo	Clave Principal	Tipo
idContenidos	Si	Entero
recurso	No	Entero
nivel	No	Entero
contenidoSuperior	No	Entero
creado	No	Fecha
modificado	No	Fecha
descripcion	No	Texto
texto	No	Texto

Tabla 4.5: Tabla Contenidos

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

Campo	Clave Principal	Tipo
idNivel	Si	Entero
nivel	No	Texto
descripcion	No	Texto

Tabla 4.6: Tabla Nivel

Por otra parte, para que pueda existir un recurso es necesario saber sobre qué contenido trata, y para ello tenemos la tabla Contenidos (ver Tabla 4.5). Esta tabla define los contenidos y su estructura jerárquica, por ejemplo, para hablar de una sala de un museo será necesario previamente albergarla dentro de una planta del mismo.

Esta tabla se compone de dos campos de texto, otros dos de tipo fecha y tres más de tipo entero, donde contenidoSuperior hace referencia a otra entrada de la misma tabla que está en un nivel jerárquico superior, si tuviera cero en ese campo significaría que no depende de otro contenido.

La única relación que esta tabla posee es con la tabla Recursos, la cual hemos explicado en la parte donde se habla de esta última tabla.

Debemos conocer que nivel de dificultad posee cada recurso que tenemos almacenado en la base de datos, siendo la tabla Nivel la que realiza esta función por medio de los tres campos que posee (ver Tabla 4.6), dos de tipo texto y otro de tipo entero, siendo esa la única relación que esta tabla tiene.

Campo	Clave Principal	Tipo
idTipoContenido	Si	Entero
tipo	No	Texto
descripcion	No	Texto

Tabla 4.7: Tabla TipoContenido

Algo similar ocurre con la tabla TipoContenido (ver Tabla 4.7), que nos dictamina el tipo de recurso de cada fichero almacenado en el servidor, cuya ruta de enlace se puede conocer a

través de la base de datos. Posee tres campos únicamente, dos de texto y otro de tipo entero para la clave principal. Posee una relación con la tabla recursos solamente.

A continuación explicaremos el uso que se da a la tabla Visualizaciones, que posee un campo fecha y tres campos enteros (ver Tabla 4.8), donde usuario y contenido indican el idUsuario e idRecursos de la persona y el recurso que se vio respectivamente, siendo estas sus únicas relaciones.

Campo	Clave Principal	Tipo
idVisualizaciones	Si	Entero
fecha	No	Fecha
usuario	No	Entero
contenido	No	Entero

Tabla 4.8: Tabla Visualizaciones

Como su propio nombre indica, nos sirve para alojar la información necesaria para conocer que usuarios han visto que recursos, así como la fecha en que se realizó.

Por último, la tabla VotacionContenido (ver Tabla 4.9) alberga los diferentes votos que los usuarios han ido dejando sobre los contenidos que han visto. Está formada por un campo fecha y otros cuatro campos enteros, donde contenido y usuario nos indican la clave principal de las tablas Recursos y Usuarios con que se relacionan.

Campo	Clave Principal	Tipo
idVotacionContenido	Si	Entero
contenido	No	Entero
votacion	No	Entero
usuario	No	Entero
fecha	No	Fecha

Tabla 4.9: Tabla VotacionContenido

4.6 Aplicación en Dispositivo Móvil

Para entender mejor el funcionamiento de la aplicación en *smartphones* mostraremos los diferentes diagramas de bloques que cada clase Java sigue, además en el capítulo siguiente mostraremos algunos *layout*² que se han implementado así como los *Views* que se utilizan en cada clase.

Antes de seguir, destacaremos varias cosas, para empezar, adelantar que una aplicación Android puede ser de varios tipos: *Activity*, es el más utilizado y nos permite representar elementos *Views* en la interfaz de usuario; *Service*, código ejecutado en segundo plano sin necesidad de interfaz; *Broadcast Receiver*, usado cuando una aplicación desea recibir y responder a un evento global como por ejemplo una llamada de teléfono; o *Content Provider*, que gestiona datos y los muestra a otras aplicaciones ejecutadas en el entorno de Android.

Nuestra aplicación estará compuesta por diferentes clases que heredan de *Activity*, siendo todas ellas independientes, además estas clases serán llamadas desde otros *Activity*, que podrán compartir información entre ellas a través de objetos de la clase *Bundle*.

Dicho esto, empezaremos hablando de la clase que se ejecuta al iniciar la aplicación, *Inicio.java*, cuyo diagrama se puede apreciar en la Figura 4.14, su función es dar la bienvenida al usuario e informarle que en todo momento va a ser necesario que disponga de conexión a internet.

Utiliza el *layout inicio.xml*, donde tenemos un *ScrollView* que contiene un *LinearLayout* con dos *TextView*, un *ImageView* y dos *Button*.

Así pues, cuando la aplicación comienza carga la clase Inicio, y esta a su vez realiza las labores de inicialización (inicializa las variables, define el *layout* que va a ser usado, inicializa los *Views*, define los oyentes de los *Views*, asocia los *Views* con el *layout* y carga los datos necesarios), esperando a que el usuario seleccione uno de los dos botones.

Si se selecciona el segundo botón abandonamos la aplicación, por el contrario, si el botón

²Los *layout* permiten al programador separar la lógica del programa frente a la interfaz de usuario. En ellos se definen los elementos y la estructura de los objetos *Views*.

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

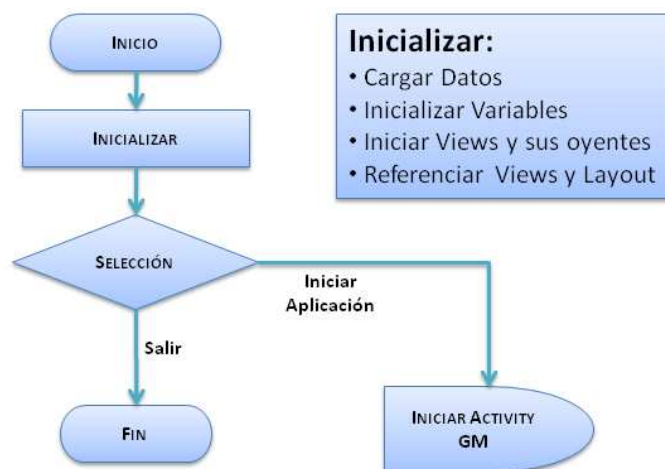


Figura 4.14: Diagrama de flujo de la clase *Inicio.java*

seleccionado es el primero iniciaremos un nuevo *Activity*, que cargará la clase *GM.java*.

Cada vez que se quiere iniciar una nueva clase, y por consiguiente un nuevo *Activity*, se realizan varios pasos: creamos un nuevo objeto de la clase *Intent*, después añadimos datos extra por medio de pares clave-valor, para a continuación iniciar el *Activity* a la espera de resultado.

La siguiente clase que vamos a tratar es *GM.java* (ver Figura 4.17). En ella se encuentra el menú principal de la aplicación, a través del cual podemos acceder a los contenidos que se ofrecen, iniciar el menú de usuario, consultar la ayuda, cambiar de idioma y abandonar la aplicación.

El modo de funcionamiento de esta clase es sencillo, una vez inicializado todos los datos necesarios se comprueba la identidad del usuario, para después esperar a la pulsación de un botón por parte del usuario. Si el usuario pulsa alguno de los cuatro primeros botones se iniciará el *Activity* correspondiente, *Contenido1.java*, *Usuarios.java*, *Idioma.java* o *Ayuda.java* respectivamente; si se pulsara el botón salir se le pedirá confirmación al usuario acerca de abandonar la aplicación.

Los datos correspondientes al usuario, nombre de usuario y contraseña, son almacenados por medio de *SharedPreferences*, lo que dota de seguridad a esta información debido a que sólo podemos acceder a ellos desde la aplicación.

Cuando se comprueba la identidad del usuario (ver Figura 4.15) se recogen los datos almace-

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

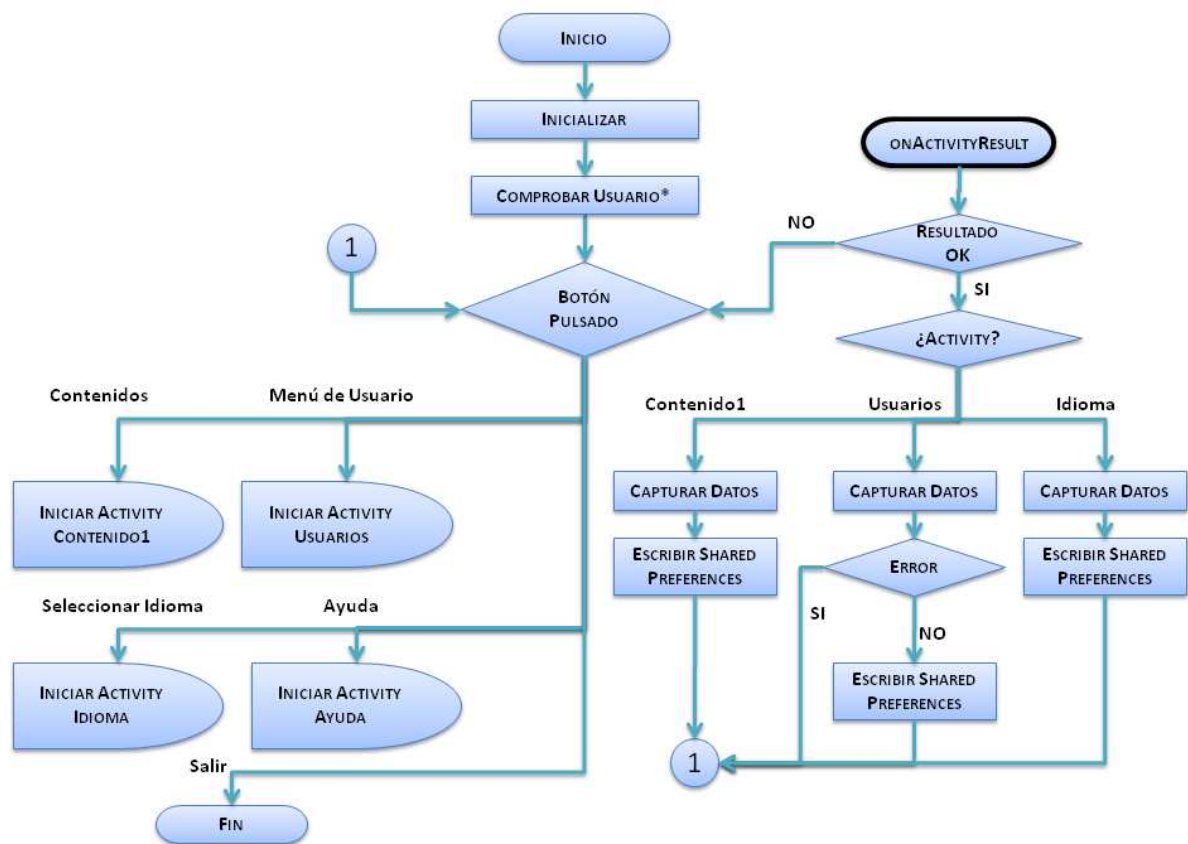


Figura 4.15: Diagrama de flujo de la clase *GM.java*

nados en *SharedPreferences* y se realiza una consulta a la base de datos sobre si esas credenciales son correctas. Si la respuesta nos devuelve algún error significa que no se tienen los datos almacenados acerca de un usuario registrado, y por lo tanto no se inicia sesión, ya que no es indispensable para poder utilizar la aplicación, sino que es únicamente obligatorio si queremos puntuar o comentar algún contenido.

Si el resultado de la consulta a la base de datos es correcta significa que el usuario está registrado y ha confirmado su cuenta, con lo que utilizaremos la información almacenada para crear nuevos comentarios, puntuaciones y almacenar información sobre contenidos vistos. Además informaremos al usuario acerca del inicio de sesión permitiéndole modificar los datos en caso de ser necesario.

En este punto vamos a realizar un nuevo inciso, en este caso para hablar sobre la clase

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

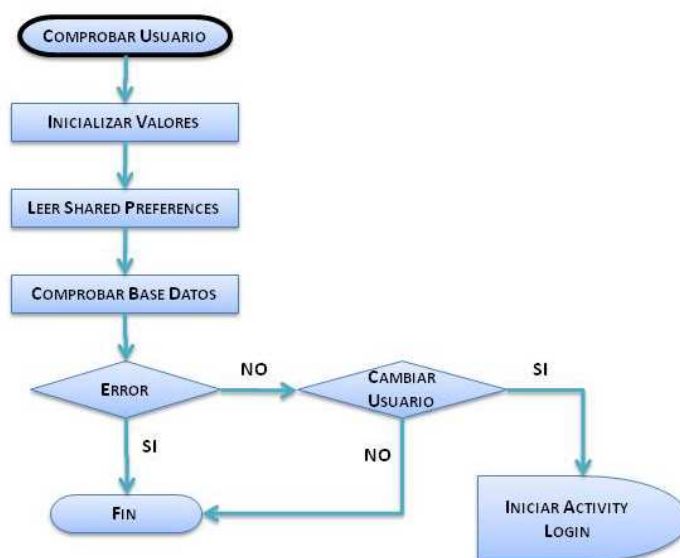


Figura 4.16: Comprobación de usuario en la clase *GM.java*

Conexión .java (ver Figura 4.17). Es la que más veces se utiliza a lo largo de la ejecución de la aplicación, puesto que es la encargada de realizar las peticiones al servidor por medio de conexiones HTTP, además captura la información en formato XML que nos devuelve el servidor, parseando dicha información para extraer los datos que esperamos en cada petición.

Por su parte, la clase *Contenido1.java* nos va a permitir seleccionar el contenido que queremos ver entre todos los disponibles por medio de listas desplegables, denominados *Spinners* en Android, o volver al menú anterior.

Su modo de funcionamiento consiste en inicializar los datos necesarios para después esperar a que el usuario haga su selección. Si pulsa el botón Volver la aplicación finalizará el *Activity* actual; en cambio si selecciona un dato de las listas desplegables se le preguntará si quiere continuar buscando o prefiere ver la información referente a ese contenido.

Al continuar la búsqueda se mostrarán todos los *Spinners* referentes a contenidos de nivel superior al seleccionado así como los contenidos del nivel inmediatamente inferior, en caso de existir.

Al seleccionar ver el contenido cargaremos la clase *Contenido.java* (ver Figura 4.19), que permite a la persona que está utilizando la aplicación acceder a los diferentes recursos disponible

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA



Figura 4.17: Diagrama de flujo de la clase Conexión.java

para ese contenido, así como consultar los comentarios y votaciones realizados por los usuarios o consultar la información sobre dicho contenido.

Cuando comienza la clase *Contenido.java* se inicializan los datos que esta clase necesita, para después esperar a que el usuario haga su selección en la interfaz. Existen cuatro opciones: volver, que terminará el *Activity* actual; ver información, que iniciará el *Activity* Información.java; seleccionar un comentario, que mostrará una lista de los comentarios introducidos por los usuarios, donde al elegir uno de ellos se cargará la clase *Comentario.java*; o seleccionar un *ImageButton*, que mostrará los recursos disponibles para ese tipo de recurso (texto, imagen, audio, vídeo o signoguía), iniciando la clase *Reproducir.java* al seleccionar uno de los mostrados.

Por su parte, al finalizar uno de esos *Activity*, se comprobará si el resultado ha sido favorable y el *Activity* terminado es *Contenido.java*. Si no es así se pasará a esperar una nueva selección del usuario, por el contrario si se cumplen ambas condiciones preguntaremos al usuario si quiere introducir un comentario, abriendo la clase *Puntuar.java* si así lo desea.

La siguiente clase que trataremos será Información.java, que se encarga de mostrar los datos almacenados en el campo texto de la tabla contenidos, acercando al usuario una sencilla explicación sobre el contenido seleccionado.

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

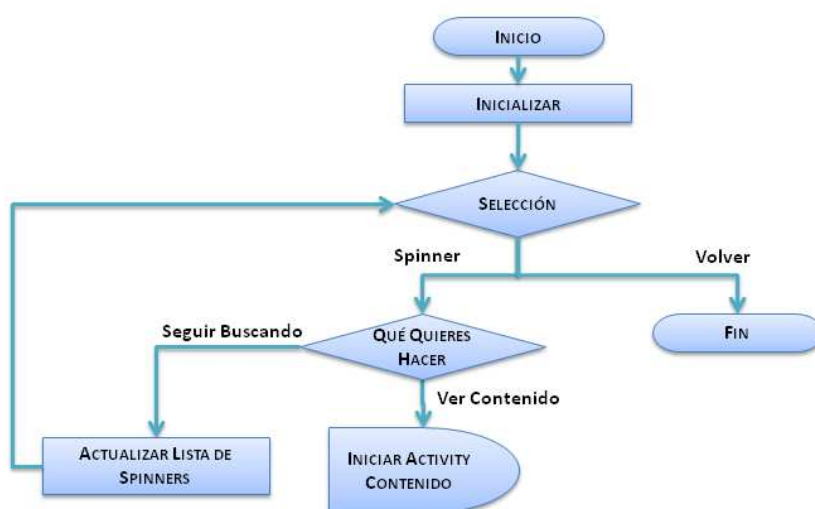


Figura 4.18: Diagrama de flujo de la clase *Contenido1.java*

Además, como se puede apreciar en la Figura 4.20, su funcionamiento consiste en inicializar los datos necesarios, mostrando al usuario la información acerca del contenido, para más tarde esperar a que éste pulse el botón volver, lo que dará por finalizado el *Activity*.

Otro *Activity* al que se accede desde la clase *Contenido.java* es *Comentario.java*, que nos mostrará la información que un usuario ha introducido sobre un recurso visto, así como la puntuación que dejó.

Para comprender su funcionamiento podemos observar la Figura 4.20, donde se aprecia como se cargan los datos necesarios en este *Activity*, para después esperar a que el usuario pulse el botón volver, lo que finalizará la aplicación.

La clase *Reproducir.java*, como su propio nombre indica, nos permite ver los archivos multimedia alojados en el servidor, ya sean de un tipo o de otro, realizando *streaming* cuando sea necesario. Este *Activity* utiliza diferentes *layout* en función del contenido seleccionado.

■ Texto

Usaremos el *layout reproducirtexto.xml*, que contiene un *ScrollView* con un *LinearLayout* dentro, que a su vez tiene tres *TextView* y un *Button*.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

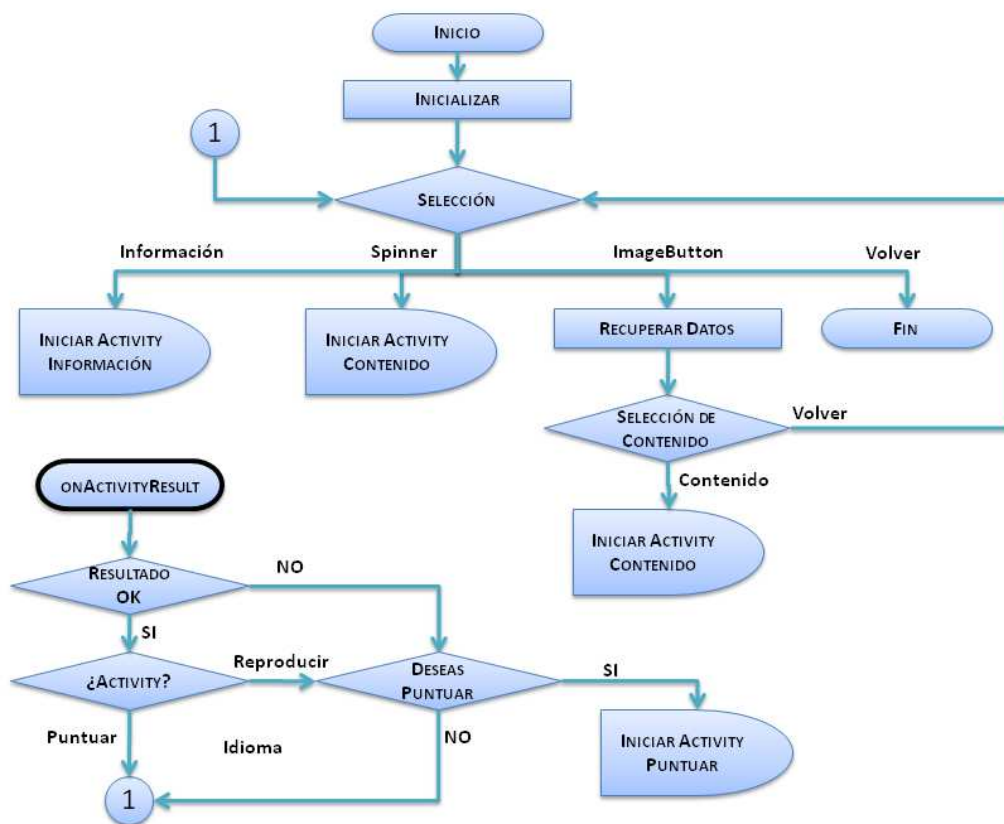


Figura 4.19: Diagrama de flujo de la clase *Contenido.java*

- Imagen

En este caso cargamos el *layout reproducirimagen.xml*, que posee un *ScrollView* conteniendo un *LinearLayout*, que a su vez tiene dos *TextView* y un *ImageView*.

- Audio

Para reproducir sonidos el *layout* utilizado es *reproduciraudio.xml*, que tiene un *ScrollView* con un *LinearLayout*, que a su vez tiene dos *TextView*, un *SeekBar* y otro *LinearLayout* horizontal conteniendo a cuatro *Button*.

- Vídeo

Este *layout*, *reproducirvideo.xml*, es el más sencillo de los cuatro, y contiene únicamente un *LinearLayout* vertical con un *VideoView* en su interior.

Su funcionamiento sigue el esquema mostrado en la Figura 4.21. Una vez inicializados los

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL



Figura 4.20: Diagrama de flujo de la clase Información.java y Comentario.java

datos y con la información recogida del anterior *Activity* sobre que tipo de contenido va a reproducir, se fija el *layout* y se inicializan los diferentes *Views*, algo en lo que difiere de las otras clases.

Si el usuario desea ver un texto o una imagen, el sistema cargará los datos del servidor y se los mostrará por pantalla, esperando a que éste seleccione volver, ya sea a través del botón o por medio de la tecla *back* del teléfono, finalizando de esta manera el *Activity*.

En otro línea muy distinta actúa esta clase cuando se ha elegido un recurso de tipo audio, ya que deberemos dotar a la aplicación de controles multimedia para un correcto uso. Una vez definido el *layout* e inicializado los datos necesarios, creamos un objeto de la clase *SeekBar*, que nos va a permitir mostrar el progreso del audio; además, generaremos un nuevo hilo que actualizará esa barra cada segundo; y por último iniciaremos la reproducción.

Por otro lado, existirán cinco controles diferentes:

- Reproducir

Si el reproductor se encuentra en modo pausa continua la reproducción, si se encuentra parado inicia el audio, pero si se encuentra funcionando no hace nada.

- Pausar

Pausa la reproducción.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

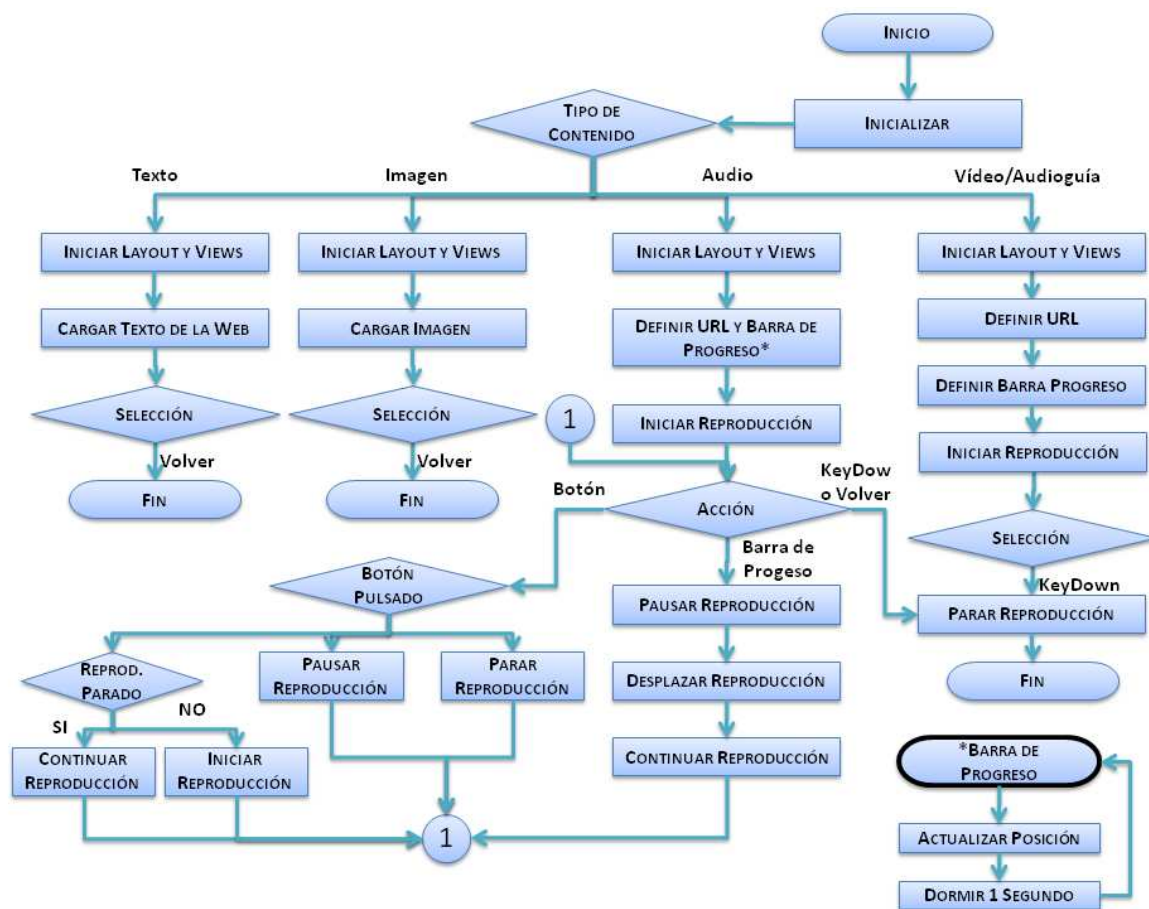


Figura 4.21: Diagrama de flujo de la clase *Reproducir.java*

- Parar

Para la reproducción.

- Volver

Detiene la reproducción y finaliza el *Activity*.

- Barra de progreso

Cuando se selecciona una posición en la barra de progreso se pausa la reproducción, a continuación se desplaza la misma al punto seleccionado en la barra, y se continua la reproducción.

El último tipo de archivo que se puede reproducir es el vídeo, ya sea por seleccionar una audioguía o un vídeo. Los pasos a seguir en este punto del diagrama son inicializar los datos necesarios, definir la URL donde se encuentra el archivo, definir una barra de progreso para poder tener

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

controles multimedia y comenzar la reproducción. El usuario podrá utilizar la barra de progreso para manipular la reproducción, o pulsar el botón volver para detener la reproducción y finalizar el *Activity*.

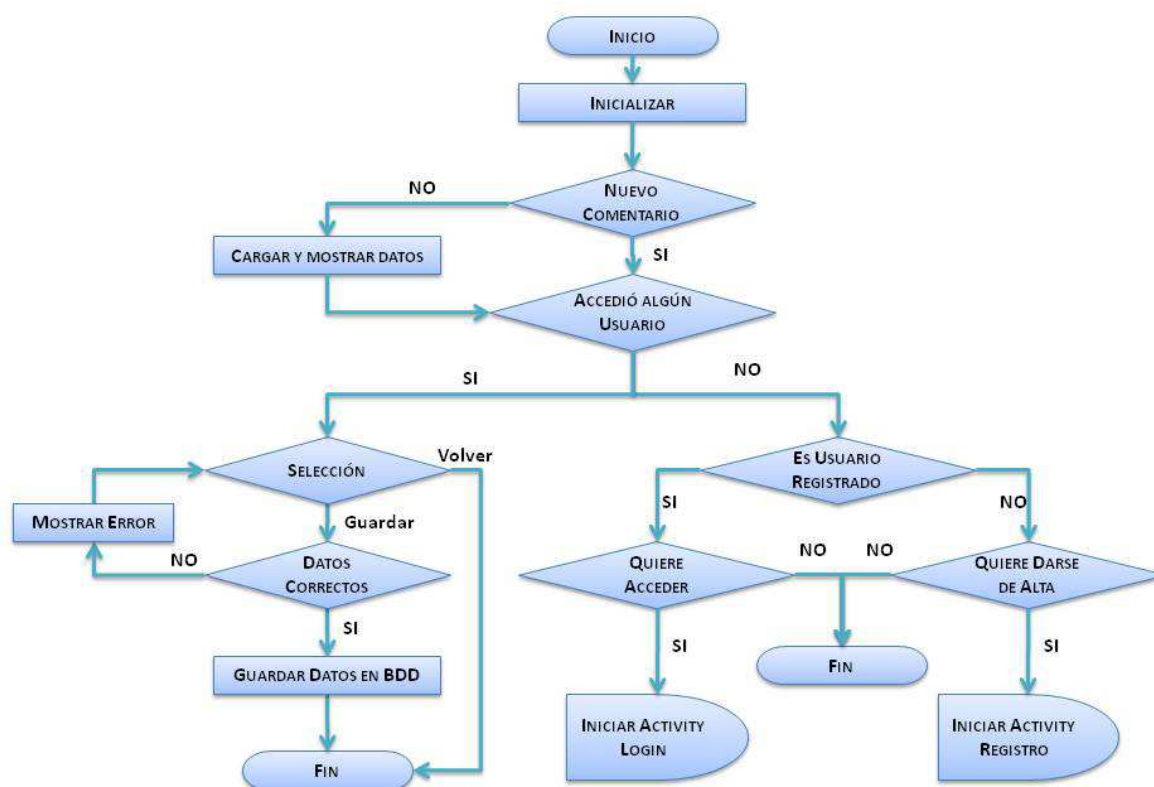


Figura 4.22: Diagrama de flujo de la clase *Puntuar.java*

Una vez el usuario ha visto el recurso y vuelto al *Activity* anterior, *Contenido.java*, se le pregunta si desea añadir un comentario, momento en el que se cargará la clase *Puntuar.java* en caso de responder afirmativamente. Esta clase se encarga de permitir al usuario guardar un nuevo comentario y puntuación sobre el recurso que ha visto, o modificar uno anterior que ya había hecho.

En la Figura 4.22 se describe su modo de funcionamiento, que como ocurría con otros *Activity*, definimos el *layout* a utilizar e inicializamos los datos necesarios. A continuación comprobamos en nuestra base de datos si ya existe ese comentario, cargando los datos si finalmente es así; después comprobamos si el usuario ha introducido sus credenciales.

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

En caso de no haberse registrado antes, se le preguntará si posee una cuenta de usuario en la aplicación, al responder afirmativamente se le volverá a preguntar si quiere acceder a su cuenta, iniciando la clase *Login.java* o finalizando el *Activity* actual si no desea hacerlo. Por otro lado, si no es usuario registrado se le preguntará si quiere serlo, iniciando la clase *Registro.java* o finalizando el *Activity*.

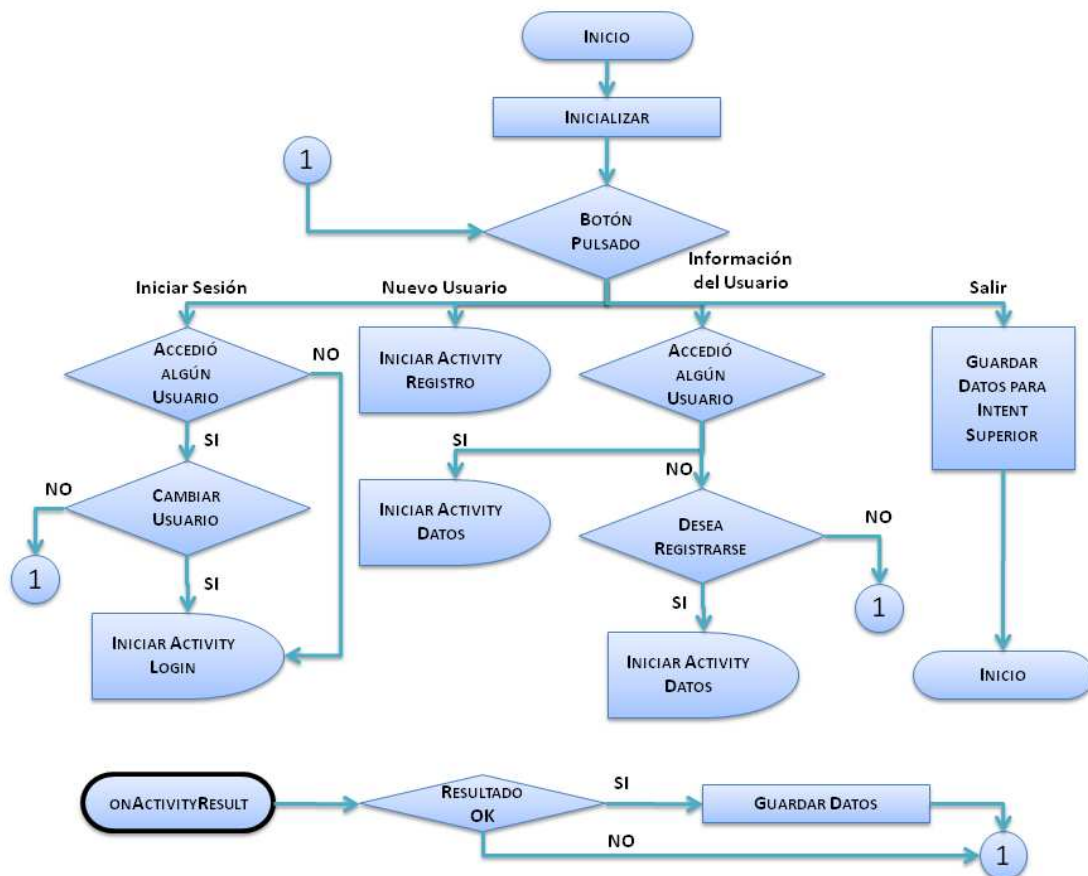


Figura 4.23: Diagrama de flujo de la clase *Usuarios.java*

Por el contrario, si ya había iniciado la sesión anteriormente se le mostrará un formulario donde introducir su comentario y puntuar el contenido, para más tarde poder elegir entre guardar o volver. El primer caso revisará si todo está correcto, mostrando un mensaje de error al encontrar un fallo, o almacenando la información en la base de datos si todo ha ido correctamente, acabando con el cierre del *Activity*. El segundo caso únicamente cierra el *Activity* actual.

Volviendo al menú principal, cuando pulsamos la tecla de acceso al menú de usuario se carga la clase *Usuarios.java*, cuyo diagrama de bloques se representa en la Figura 4.23. Desempeña

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

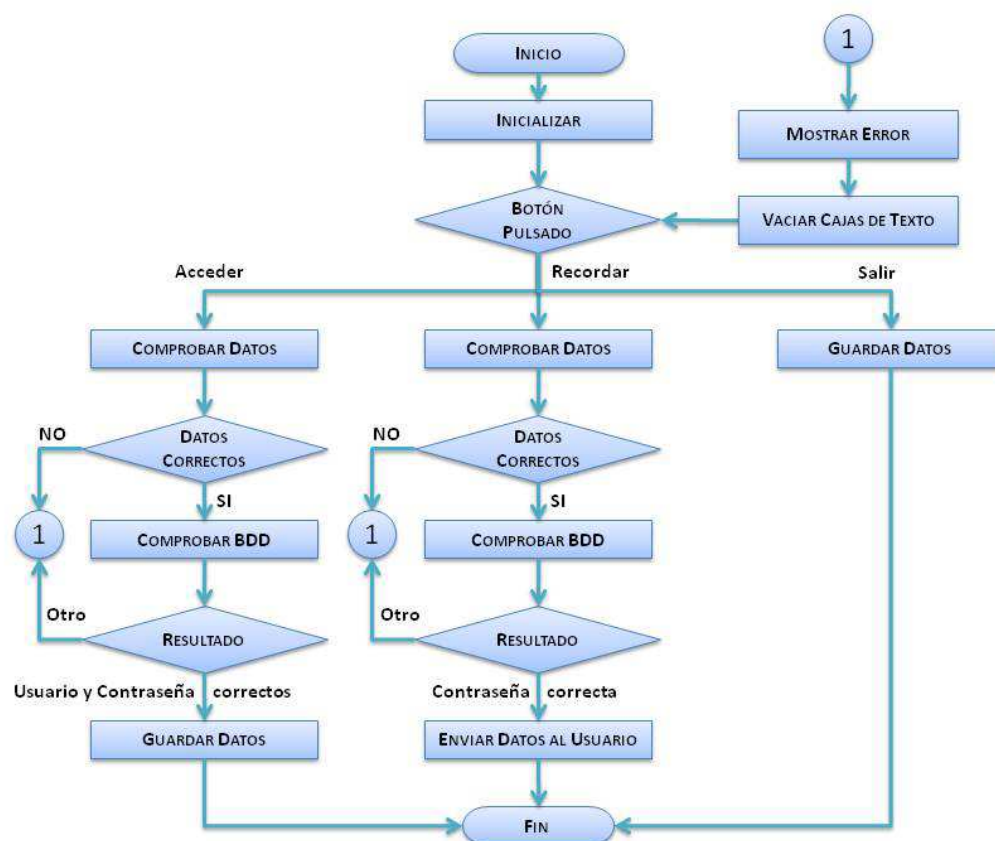


Figura 4.24: Diagrama de flujo de la clase *Login.java*

la función de proporcionar al usuario diferentes opciones asociadas a su cuenta privada en la aplicación.

Una vez inicializados los datos necesarios y definido el *layout* que mostraremos al usuario, se espera a que el usuario pulse un botón. Si el usuario desea volver se guardarán los datos y finalizará el *Activity*. Si desea iniciar la sesión se comprobará si lo ha hecho antes, de ser así se le preguntará acerca de cambiar el usuario actual, tanto si desea cambiar el usuario actual como si quiere iniciar sesión se comenzará el *Activity* de la clase *Login.java*; si no desea cambiar la sesión no se hará nada. Cuando la selección es un nuevo registro automáticamente se inicia la clase *Registro.java*. En cambio, al pulsar sobre los datos personales se comprueba si se ha iniciado sesión, comenzando la clase *Datos.java* en caso de haber sido así.

Su manera de actuar aparece esquematizada en la Figura 4.24, donde vemos que una vez

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

se ha iniciado los valores que se necesitan se espera a que el usuario introduzca los datos en el formulario y pulse algún botón. Si pulsa el botón volver finalizará el *Activity*. Al pulsar el botón se comprueba que los datos introducidos son correctos, de ser así se envían a la base de datos para comprobar la identidad, guardando los datos de las credenciales de usuario y finalizando el *Activity* si la respuesta es afirmativa. Si se produjese algún error en la comprobación de la base de datos o por no haber introducido los campos de forma adecuada se mostrará un mensaje de error.

Cuando se quiere recordar la contraseña, realizamos unos pasos similares a iniciar sesión, salvo que ahora sólo comprobamos si el usuario existe, enviándole un correo con los datos de la cuenta al correo electrónico asociado, finalizando el *Activity*.

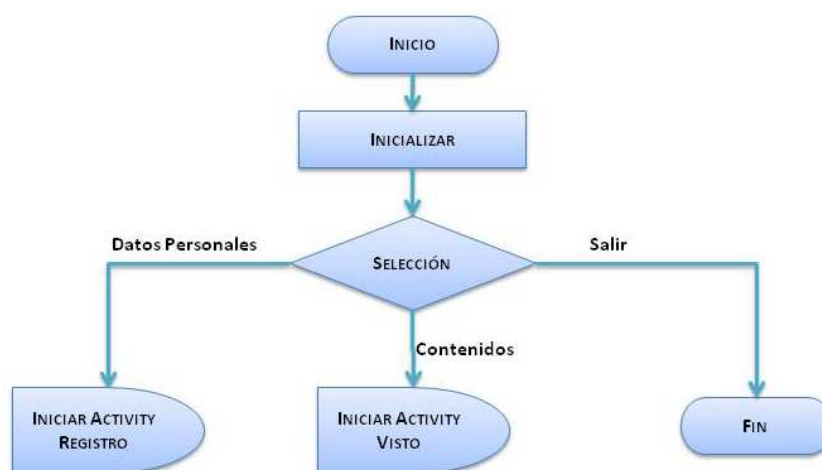


Figura 4.25: Diagrama de flujo de la clase *Datos.java*

En cuanto a la clase *Datos.java*, comentar que nos ofrece un menú para los usuarios registrados que quieran acceder a sus datos personales para modificarlos, o deseen conocer los comentarios que han realizado.

Su funcionamiento se detalla en la Figura 4.25, consiste en inicializar los datos necesarios para acto seguido esperar a la selección por parte del usuario. Si bien elige datos personales se cargará la clase *Registro.java*, si bien selecciona contenidos vistos se iniciará la clase *Vistos.java*. Al pulsar sobre el botón volver se cerrará el *Activity* volviendo al anterior.

Por su parte la clase *Visto.java* nos permite ver aquellos comentarios y puntuaciones que

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

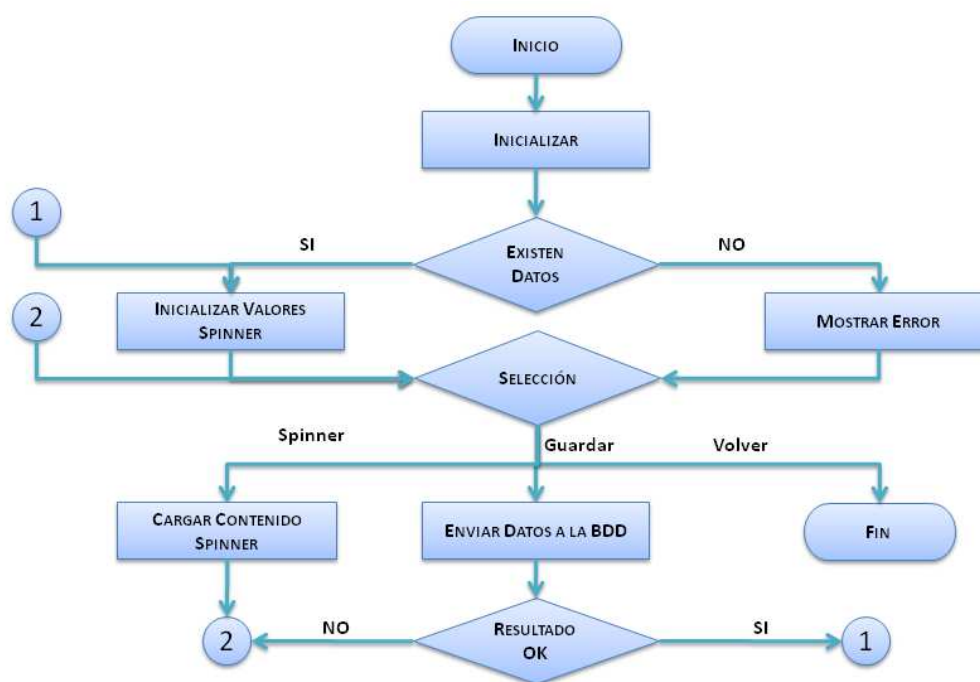


Figura 4.26: Diagrama de flujo de la clase *Visto.java*

se han realizado a los contenidos que hemos visto.

Como vemos en la Figura 4.26, una vez iniciados los datos necesarios, comprobamos si el usuario ha realizado algún comentario, notificando en caso de no poseer ninguno, o cargando la información de la base de datos en los *Spinners* si hubiera alguno.

Cuando el usuario seleccione algún comentario del *Spinner* se mostrará en el formulario la información de dicho comentario, así como la puntuación correspondiente. Si desea modificar algún campo, podrá guardar ese cambio en la base de datos a través del botón correspondiente, abandonando el *Activity* al pulsar el botón volver.

La última opción dentro del menú de usuario es realizar un nuevo registro, labor que lleva a cabo la clase *Registro.java*. Para conocer cómo funciona esta clase observaremos la Figura 4.27, donde apreciamos que una vez se ha inicializado, el primer paso consiste en comprobar si se nos han pasado datos mediante *Intent*, señal inequívoca de que mostraremos la información de un usuario, rellenando los campos del formulario con los que la base de datos tiene de esa persona.

Si pulsamos el botón enviar, comprobaremos que todos los datos han sido introducidos de

CAPÍTULO 4. IMPLEMENTACIÓN DEL SISTEMA

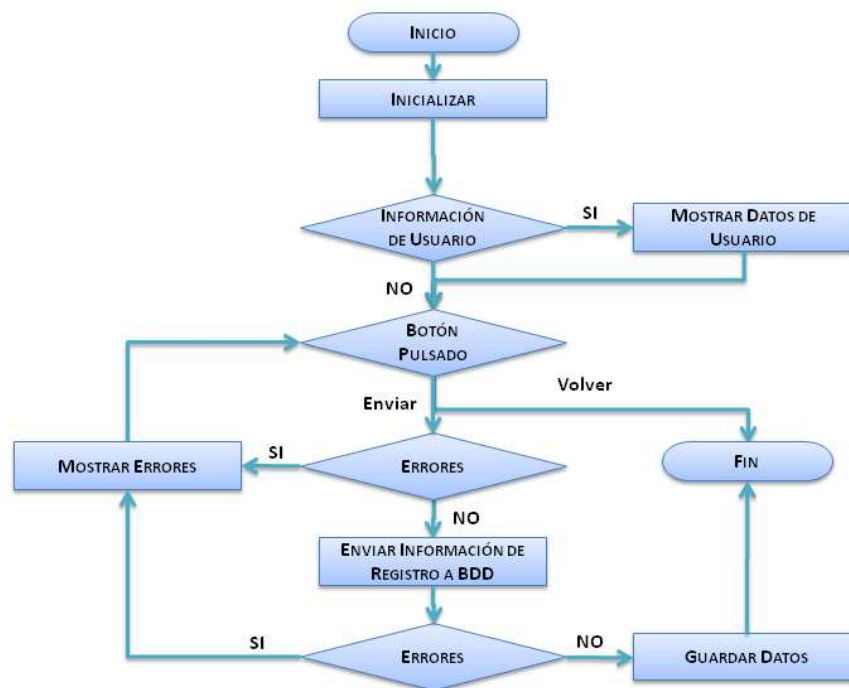


Figura 4.27: Diagrama de flujo de la clase *Registro.java*

manera correcta, mostrando los errores de no ser así. Si se han introducido de forma adecuada se enviará la información a la base de datos para que registre los cambios en el usuario, o para que añada uno nuevo a la misma.

El botón volver, como viene siendo habitual, nos permite cerrar el *Activity* actual, volviendo al que nos encontrábamos anteriormente.

Volviendo al menú principal de la aplicación nos encontramos con la posibilidad de cambiar el idioma. En este punto destacar que se ha implementado el funcionamiento de esta acción, aunque la aplicación no permite ese cambio, puesto que no tenemos disponibles los recursos en otros idiomas.

Al observar la Figura 4.28, vemos como tras inicializar los datos se carga en el *Spinner* la lista de idiomas disponible, pudiendo finalizar el *Activity* por medio del botón volver, o seleccionar otro idioma en el *Spinner*. Esto último guardará en las *SharedPreferences* los datos correspondientes,

4.6. APLICACIÓN EN DISPOSITIVO MÓVIL

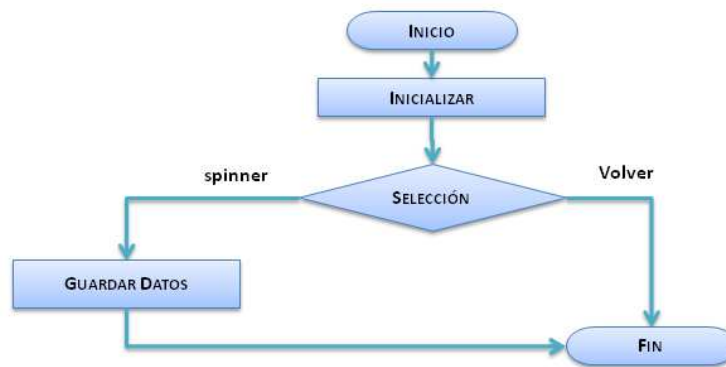


Figura 4.28: Diagrama de flujo de la clase *Idioma.java*

de forma que la próxima vez que se inicie la aplicación utilizará el nuevo idioma.

La última clase corresponde a *Ayuda.java*, la cual muestra un texto de ayuda sobre las diferentes partes que tiene esta aplicación. Tal y como se aprecia en la Figura 4.29 Una vez inicializada la clase, y con los datos necesarios cargados el usuario puede elegir en el *Spinner* una sección de este software, mostrándole el texto correspondiente a la ayuda de dicha sección. Para finalizar el *Activity* pulsaremos el botón volver.

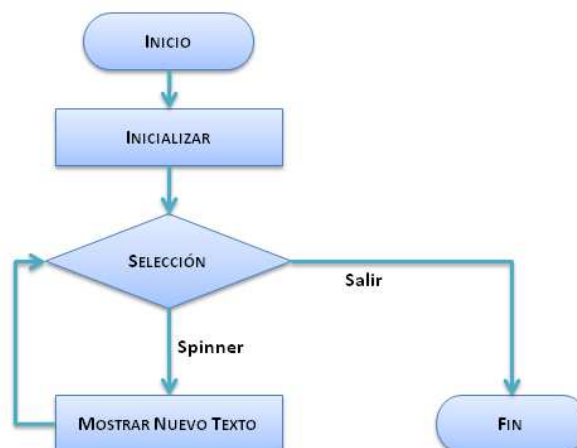


Figura 4.29: Diagrama de flujo de la clase *Ayuda.java*

4.7 Conclusiones

Decantarse por un entorno de desarrollo u otro nos va a condicionar en la resolución del proyecto que tenemos que llevar a cabo, debido a las diferencias que podemos encontrar entre sistemas similares. En nuestro caso hemos optado en todo momento por reducir los costes utilizando herramientas gratuitas que presenten funcionalidades similares a las opciones de pago.

A su vez, las herramientas utilizadas nos han ayudado en todo momento a la programación de las diferentes partes del proyecto, siendo notable la ayuda que proporcionan en labores complejas como son la depuración de errores, la visualización del sistema en la red local para comprobar el correcto funcionamiento, o la gestión de la base de datos por medio de interfaz gráfica.

Por otro lado, la elección de Android como SO sobre el que desarrollar la aplicación ha venido determinado por múltiples factores: posibilidad de realizar aplicaciones multiproceso (necesario para la barra de progreso de la reproducción de archivos de sonido), comunidad de desarrollo en esta plataforma, recursos disponible, o posibilidad de desarrollar para esta plataforma sin necesidad de adquirir licencias siendo posible hacerlo de manera gratuita (no hay que olvidar que la programación puede realizarse o bien en C/C++, o bien en Java, dos de los lenguajes más extendidos como se desprende de la Figura 2.4). Añadir que Android proporciona al programador una completa plataforma de desarrollo sin coste alguno, desde la cual puede sacarse todo el rendimiento a este SO.



5

Verificación del Sistema

Es impensable que hoy en día un producto salga al mercado sin pasar previamente un proceso de verificación, ya que es en esta etapa donde se observa la calidad del producto y se certifica que cumple con las expectativas marcadas para el mismo al inicio de su fase de creación.

CAPÍTULO 5. VERIFICACIÓN DEL SISTEMA

5.1 Introducción

En el presente capítulo detallaremos las pruebas que hemos realizado al sistema para confirmar que los diferentes requisitos enunciados en el Capítulo 3 se ven cumplidos, además utilizaremos distintas capturas de pantallas que ayudarán a su comprensión.

En primer lugar, analizaremos cuál ha sido el contexto en el que se han realizado las pruebas, escenario, equipos, tiempos; para continuar detallando los resultados obtenidos de las mismas.

Para finalizar el capítulo realizaremos un balance sobre todos los requisitos que teníamos al comienzo del desarrollo de la aplicación, así como los resultados que se han obtenido al verificarlos.

5.2 Entorno de prueba

El test de la aplicación se ha realizado utilizando un servicio de registro de dominios y alojamiento web de una empresa privada ajena al proyecto, utilizando el plan de alojamiento básico que ofrece dicha organización.

Los equipos que se han necesitado han sido un ordenador con conexión a Internet y capacidad para abrir un navegador y un teléfono inteligente *smartphone* con el SO Android instalado, concretamente hemos empleado el teléfono marca *Huawei* modelo *Ivy* con la versión 2.1 del citado SO.

Además, destacar que se dispone de acceso a Internet por medio de un router con conexión *Asymmetric Digital Subscriber Line* (ADSL) con ancho de banda de 6 Mbps, que permite, tanto al ordenador como al teléfono, conectarse por medio de WiFi.

5.3 Resultados obtenidos

Hablaremos de los resultados obtenidos, realizando una división en los mismos, por un lado la parte web de la aplicación, y por otro lado la parte que tiene que ver con el teléfono inteligente.

Identificaremos los diferentes requisitos que en cada caso se verifican introduciendo al final del párrafo entre corchetes el código que dimos a cada requisito en el Capítulo 3.

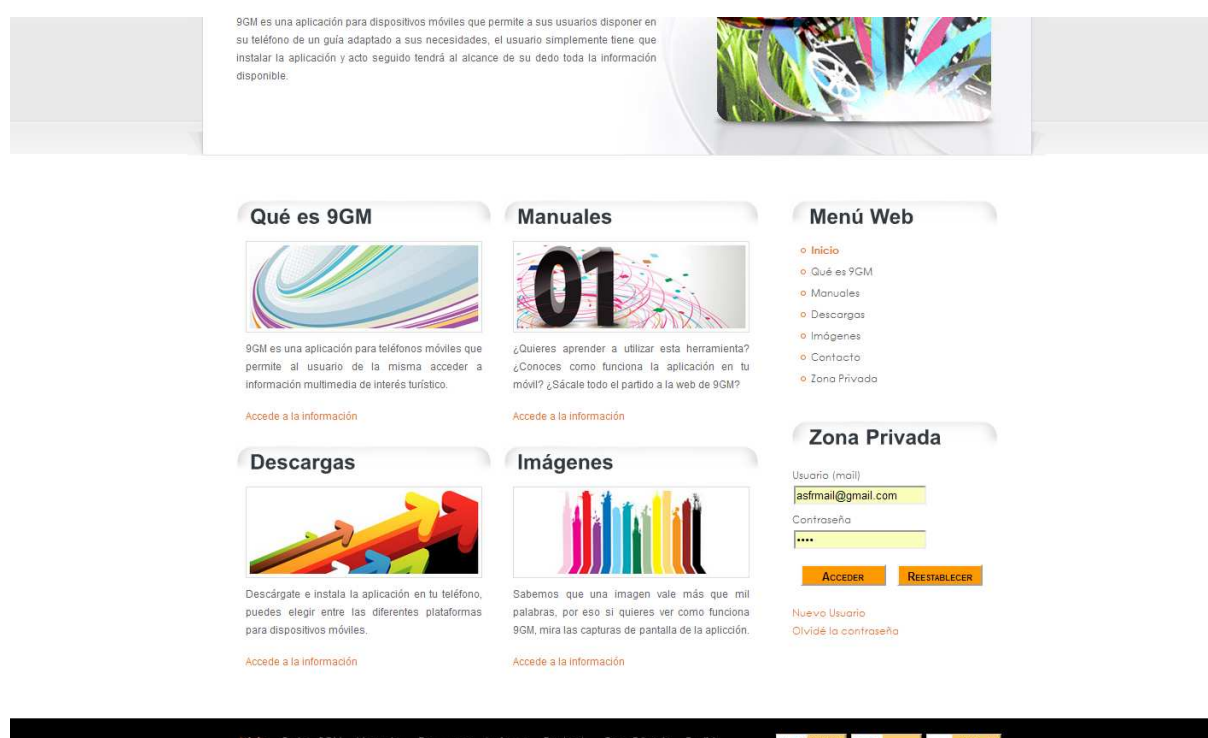


Figura 5.1: Página de inicio de la plataforma web

5.3.1 Parte Web

Una de las pruebas consistió en ver la independencia de la aplicación web frente al navegador que utilizamos normalmente, habiendo realizado la misma con los programas Internet Explorer, Mozilla Firefox, Google Chrome, Netscape y Safari, obteniendo resultados positivos. Por ello, para poder acceder a la plataforma web de esta aplicación introduciremos la

CAPÍTULO 5. VERIFICACIÓN DEL SISTEMA

dirección <http://www.infined.es/pfc/> en el navegador que deseemos. Además la resolución de pantalla recomendada para la visualización correcta de la hoja de estilos es de 1024 píxeles de ancho[RUCW001, RURW003, RURW004, RURW005, RURW006, RSFW003, RSRW004, RSIW001, RSCW002, RSCW003].

Cuando un usuario se instale el software y lo ejecute no tendrá almacenada ninguna credencial que identifique al usuario como registrado, con lo que será necesario realizar *login* para entrar a la zona privada, por ello, una vez hemos iniciado la página anterior podemos acceder a nuestra cuenta de usuario por medio del enlace que aparece en el menú de Zona Privada que podemos ver en la Figura 5.1, mismo lugar que aparece la opción de cerrar sesión si ha sido iniciada con anterioridad. Añadir que mientras no cerremos el navegador, o le demos a cerrar sesión, éste seguirá reconociéndonos de manera automática puesto que en cada carga de una nueva página verifica si se ha realizado el acceso del usuario[RUCW002, RUCW003, RURW007, RSSW001, RSSW002].

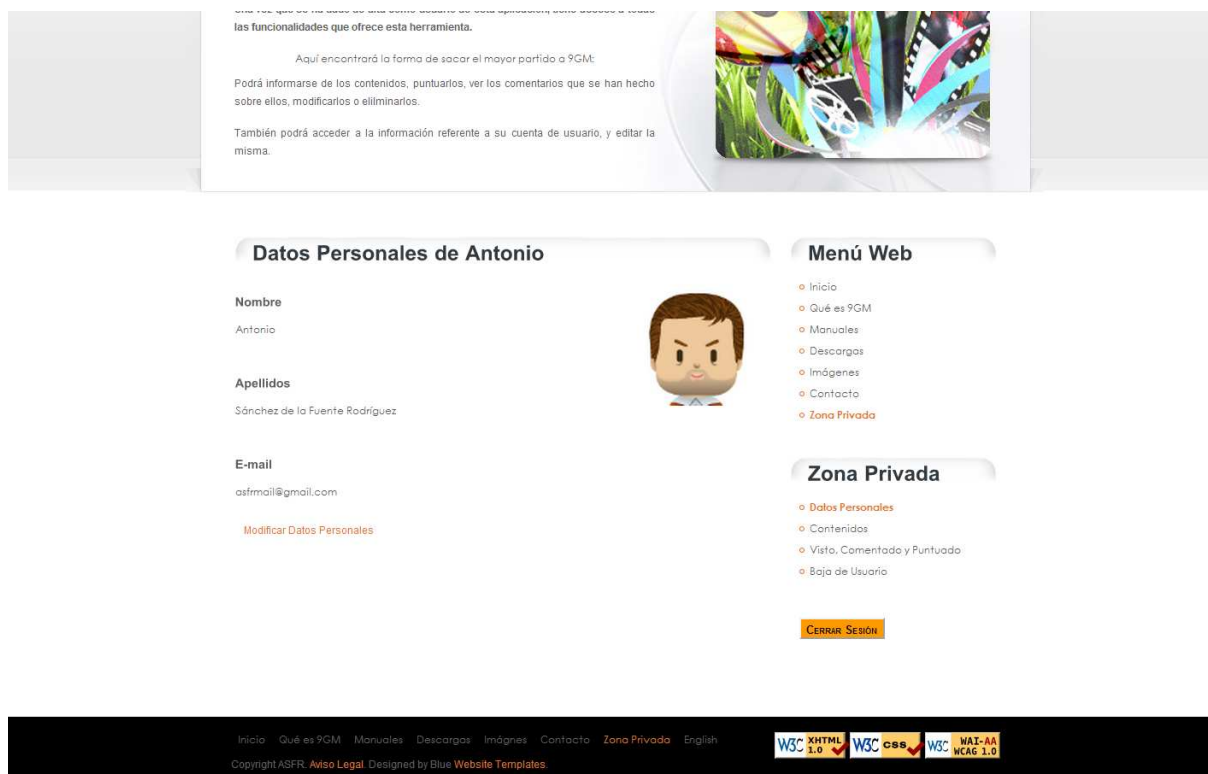


Figura 5.2: Información personal del usuario

5.3. RESULTADOS OBTENIDOS

Por otro lado, los enlaces para descargar los diferentes manuales o la aplicación para teléfonos inteligentes se encuentran activos, descargándose de forma correcta cada archivo requerido desde el alojamiento web, el cual tiene un límite de 400 MB[RUCW004, RSCW001].

Los tiempos de respuesta encontrados en la carga de cualquier página de la plataforma web, incluidas aquellas que realizan un procesamiento elevado en el lado del servidor o requieren el acceso a la base de datos, son inferiores a 2 segundos¹, tiempo suficiente para el uso de páginas web, en las que el tiempo de respuesta no es un factor crítico[RURW001, RURW002, RSRW001, RSRW002, RSRW003].

Lo primero que encuentra un usuario tras iniciar sesión son sus datos personales, como podemos ver en la Figura 5.2, pudiendo modificarlos a través del enlace que encuentra de los datos de usuario[RUCW005].

The screenshot displays a web interface with several sections:

- Campus de la Universidad**: A paragraph describing the university's commitment to quality and its role in the European Higher Education Area.
- Recursos Multimedia Disponibles**: A list of resources categorized by difficulty level:
 - Dificultad Baja - Texto - Imágenes
 - Dificultad Media - Audio - Imágenes - Signografía
 - Dificultad Alta - Imágenes - Video
- Comentarios de los Usuarios**: A list of user comments with details like username, vote, and date. It includes links to 'Ver comentario'.
- Lista de Contenidos**: A list of content items including 'Campus de la Universidad', 'Cámpus de Getafe', 'Cámpus de Leganés', 'Cámpus de Colmenarejo', 'Departamentos de la UC3M', and 'Servicios'.
- Menú Web**: A list of navigation links: Inicio, Qué es 9GM, Manuales, Descargos, Imágenes, Contacto, and Zona Privada.
- Zona Privada**: A list of user-related links: Datos Personales, Contenidos, Visto, Comentado y Puntuado, and Baja de Usuario.
- CERRAR SESIÓN**: A button to log out.

Figura 5.3: Datos mostrados sobre cada contenido

¹Se han realizado diferentes medidas del tiempo de respuesta, dando de media como resultado una velocidad de transición entre páginas inferior a los dos segundos

CAPÍTULO 5. VERIFICACIÓN DEL SISTEMA

Si por el contrario, lo que un usuario quiere hacer es modificar un comentario podrá hacerlo a través del menú Visto, Comentado, Puntuado, donde se le muestran todos los comentarios, pudiendo modificarlos a través de un formulario web[RUCW009].

Como podemos apreciar en la Figura 5.3, desde la pestaña Contenidos, en el menú de la Zona Privada, accedemos a la información almacenada en la base de datos sobre un contenido, comentarios, puntuaciones, información sobre el contenido y la selección de otros contenidos. Añadir que si pulsamos sobre un recurso multimedia, éste se iniciará para mostrarlo tal y como adelantamos en el Capítulo 4, por medio del *Activity Reproducir.java*, no presentando problemas ni de tipo ni de formato.[RUCW006, RUCW007, RUCW008, RSFW001, RFSW002].

Por último, los datos necesarios son almacenados en el alojamiento web antes mencionado, lugar que albergará los archivos multimedia utilizados, la base de datos y la información almacenada en ella, y los archivos correspondientes al portal web[RSIW002].



Figura 5.4: Autologin y acceso de usuarios

5.3. RESULTADOS OBTENIDOS



Figura 5.5: Capturas sobre Ayuda y Salir

5.3.2 Parte Móvil

Empezaremos destacando que la aplicación ocupa 380 KB de espacio de memoria en el teléfono inteligente, elemento indispensable para poder ejecutar la aplicación, así como que el mismo tenga instalado el SO Android con soporte para nivel 3 de la API y conexión a internet. Por otro lado la resolución de la pantalla no va a suponer un límite en los modelos compatibles con la aplicación debido a que en los diferentes *layout* no se especifican medidas para la altura ni el ancho[RURM004, RURM005, RURM006, RSRM004, RSIM001, RSCM002, RSCM003, RSCM004].

El primer paso para disfrutar de la aplicación en Android es instalarla, algo tan sencillo como descargar de la web al móvil y pulsar sobre el archivo, lo que hará que el sistema lo instale de manera automática. Ahora podemos usar el icono para iniciar la aplicación[RUCM001, RUCM002, RSFM002, RSFM003].

Una vez iniciada la aplicación se leerán las credenciales del usuario almacenadas en las

CAPÍTULO 5. VERIFICACIÓN DEL SISTEMA

SharedPreferences (ver Figura 5.4), con lo que será recordado si antes se ha dado de alta, además el usuario puede acceder a través de la opción del menú de usuario, realizándose de forma automática el logout del anterior usuario y guardándose los nuevos datos de usuario (ver Figura 5.4). Los únicos parámetros iniciales son aquellos comentados antes y que son almacenados en las *SharedPreferences*, siendo los únicos datos modificables[RURM007, RURM008, RURM009, RUCM005, RUCM006, RSFM005, RSFM006, RSIM002, RSSM001, RSSM002].

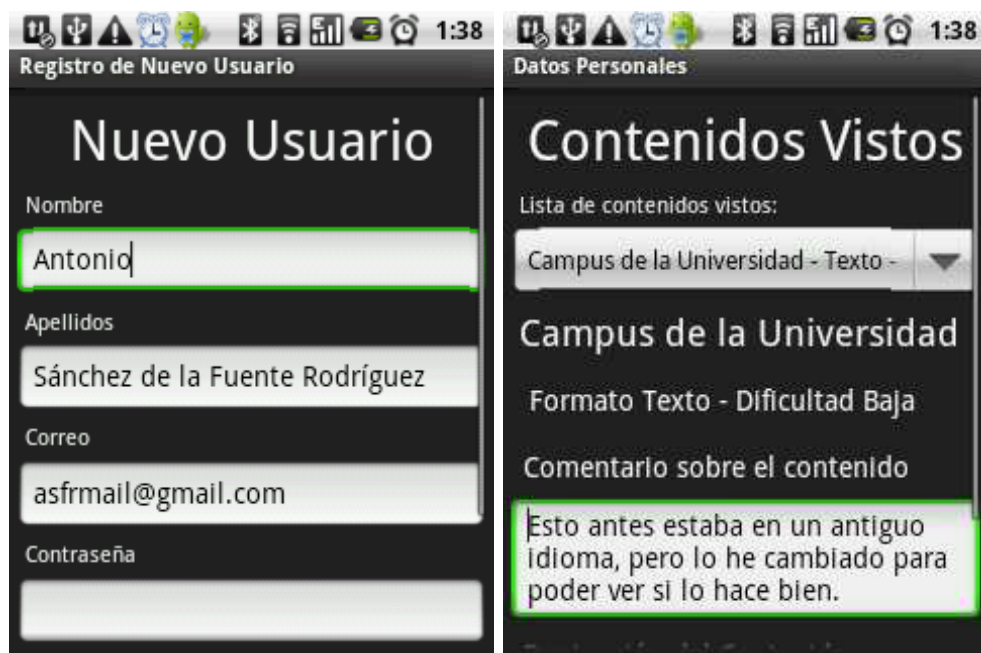


Figura 5.6: Datos Personales y Modificar Comentarios

Por su parte, el tiempo de respuesta de los diferentes *Activity* vendrá determinado por la velocidad de la conexión a internet. Si no existen problemas en el acceso a esta red, el tiempo de carga de la aplicación es inferior a los dos segundos² presentan un tiempo de carga similar, así como la captura de los datos almacenados en el servidor[RURM001, RURM002, RURM003, RSRM001, RSRM002, RSRM003].

Entre las opciones que posee el usuario desde el menú de inicio destacaremos el acceso a los contenidos, acceso al menú de usuario, la posibilidad de ver la ayuda en referencia a los contenidos (ver Figura 5.5), o salir de la aplicación (ver Figura 5.5), además de seleccionar el

²Para llegar a estos datos se ha calculado la media del tiempo que tarda la aplicación en las transiciones entre *Views*.

5.3. RESULTADOS OBTENIDOS

idioma que desee[RUCM003, RUCM004].

El menú de usuario permite realizar varias gestiones: acceder a la cuenta, registrarse o consultar la información sobre el usuario actual, de haberlo. Si queremos modificar un comentario que hemos dejado antes, usaremos la opción disponible en el menú de datos personales (ver Figura 5.6)[RUCM010].



Figura 5.7: Selección y Menú de Contenido

Sin embargo, la zona de mayor importancia en la aplicación son los contenidos (ver Figura 5.7), puesto que es allí donde el usuario puede consultar todos los recursos disponibles, así como los comentarios introducidos. Lo primero que el usuario puede realizar es seleccionar el contenido que desea ver, para más tarde elegir que desea ver[RUCM007].

Una vez seleccionado el contenido las siguientes posibilidades que nos ofrece el software son elegir un recurso, información sobre el contenido (ver Figura 5.8) o ver los comentarios de otros usuarios(ver Figura 5.8). Si deseamos reproducir un contenido se abrirá una nueva ventana mostrando o reproduciendo el mismo(ver Figura 5.9)[RUCM008, RUCM009, RSFM001, RSFM004].

CAPÍTULO 5. VERIFICACIÓN DEL SISTEMA



Figura 5.8: Información y Comentarios sobre los Contenidos

Una vez reproducido el recurso se nos preguntará si queremos dejar un comentario sobre el contenido que acabamos de observar (ver Figura 5.9), pudiendo elegir otro recurso, o volviendo para realizar una nueva búsqueda de contenidos[RUCM008].

5.4 Conclusiones

Una vez realizadas las comprobaciones pertinentes, obtenemos resultados favorables acerca del funcionamiento de la aplicación, verificándose todos los requisitos que al comienzo del desarrollo se fijaban.

Además, se desprenden varias conclusiones tanto de la parte web como de la móvil:

- Por un lado la aplicación web funciona de manera correcta, el tiempo de carga de las diferentes páginas es bajo, así como el empleado a la hora de realizar consultas a la base de datos, los enlaces también funcionan correctamente, y los contenidos no tienen problemas al



Figura 5.9: Reproducción y Puntuación de contenidos

presentarse en el navegador. Añadir que el uso de las *Sessions* de PHP nos proporcionan el soporte necesario para que los usuarios puedan registrarse de manera óptima. Además, los mensajes de correo electrónico son enviados y recibidos sin problemas. Tampoco presentan problemas la edición de las diferentes entradas.

- En cuanto a la aplicación Android, destacar: la agilidad en la carga de la aplicación, de los diferentes *Activity*, así como de las consultas a la base de datos junto al posterior parseo de la información XML. El tiempo de carga de los contenidos para su reproducción mediante *streaming* está próximo a los 5 segundos para vídeos de 300 kbps, aunque este dato se ve modificado por la calidad y la duración de los vídeos.

Por todo lo anterior, el funcionamiento de la aplicación desarrollada es correcto, no presentando problemas en las diferentes acciones del usuario.



III

Conclusiones del Proyecto



6

Conclusiones y Líneas Futuras

Los sistemas de información deben mantenerse siempre al día, al estar en constante evolución, haciéndose necesaria la evaluación de los sistemas, que se lleva a cabo para identificar puntos fuertes y débiles, valorando la forma en que funciona el sistema.

CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

6.1 Conclusiones

En este proyecto se ha desarrollado un prototipo de Guía Multimedia utilizando teléfonos inteligentes con el SO Android junto a un servicio de hospedaje web, que permiten utilizar la aplicación de forma remota y ubicua tanto para la administración de contenidos, como para el uso de la misma.

Las herramientas utilizadas para llevarlo a cabo cuentan con la licencia de uso GNU GPL, permitiendo a cualquier persona la accesibilidad a las mismas, y reduciendo a su vez el presupuesto del proyecto.

Se ha procurado que el diseño de la aplicación sea lo mas eficiente posible, intentando utilizar únicamente aquellos dispositivos estrictamente necesarios. Ha sido imprescindible el uso de un ordenador, un *router* con conexión inalámbrica, un teléfono inteligente *smartphone* y un dominio público con alojamiento web.

Utilizar un SO libre en el que se puede realizar el desarrollo de aplicaciones, como es Android, implica que se puedan añadir nuevas funcionalidades o modificar las que ya existen en el software.

El desarrollo de la web de la aplicación ha cuidado el cumplimiento con los estándares de programación web HTML y CSS, así como las directivas de accesibilidad.

En todo momento se ha intentado dotar al prototipo de un contexto que contase con la mayor veracidad posible, utilizando un equipo real para la consecución de las diferentes pruebas, así como un servicio de alojamiento de datos remoto para que los datos fueran enviados a través de una red de acceso público.

Los mayores problemas encontrados en el desarrollo han sido relativos a la programación en Android para la aplicación en los teléfonos inteligentes, debido al esfuerzo que supone enfrentarse a un nuevo modelo de desarrollo. Así pues, Android impone algunas restricciones que se han debido tener en cuenta a la hora de programar, como son los formatos de vídeo permitidos o los tamaños de imágenes aceptadas para su reproducción.

Además, deberemos tener en cuenta la capacidad de la red a la hora de enviar tráfico

multimedia, ya que habrá que dividir la misma entre el número de usuarios de la aplicación, con el fin de poder dotar a cada teléfono inteligente de ancho de banda suficiente.

El resultado final ha sido una herramienta que cumple con los objetivos marcados al principio del proyecto, y que da la posibilidad a los agentes turísticos de evitar adquirir un hardware propio para ofrecer a los usuarios este servicio de guía.

6.2 Líneas Futuras

Una vez realizada la aplicación prototipo que cuenta con la funcionalidad deseada, el siguiente paso es mejorar la misma por medio de nuevas opciones o modificaciones de las funcionalidades ya existentes.

Como esta aplicación es usada en equipos de última tecnología, encontramos diferentes opciones para ampliar las funciones que ofrece la misma tanto en los teléfonos inteligentes como en los dispositivos que permiten consultar webs.

Así pues podemos enumerar algunas de las posibles líneas de trabajo futuro que se pueden basar en este proyecto:

- Desarrollo en otros SO

El primer paso para generar una aplicación soportada en la mayor parte de los SO para teléfonos inteligentes se ha dado, lo siguiente puede ser implementar esta misma herramienta para iPhone, Symbian o Windows Phone.

- Mejora en la apariencia

La apariencia que una aplicación tiene puede determinar la elección de un usuario acerca de su uso, por ello creemos que un estudio de marketing acerca de los gustos y necesidades de los usuarios plasmándolos en los menús de usuario y la colocación de los enlaces ayudará a mejorar la aplicación.

- Seguridad en las conexiones

Otro punto de estudio futuro puede ser mantener conexiones seguras siempre que se trabaje

CAPÍTULO 6. CONCLUSIONES Y LÍNEAS FUTURAS

con datos privados, así como la encriptación de los mismos para evitar que puedan ser capturados para otros fines diferentes a los de esta aplicación.

- Videocomentarios

La cámara de fotos y de vídeo es una característica que traen consigo prácticamente todos los teléfonos a día de hoy, algo que puede ser usado para grabar los comentarios de los usuarios, y poder captar toda la información que es transmitida por medio de gestos.

- Capacidades de red

Otra línea de trabajo futura puede ser realizar un estudio de las necesidades en un entorno real, para que esta aplicación pueda ser utilizada con un número de usuarios elevados, por ejemplo en un museo.

- Identificación de contenidos

Para la identificación de contenidos podemos optar por métodos que se añadan a los ya existentes, como puede ser reconocer los contenidos mediante códigos alfanuméricos. Por otro lado, Android proporciona soporte para la identificación de códigos de barra, a través del cual podemos llegar al contenido requerido.

- Localización del usuario

La selección de un contenido puede realizarse también en base a la posición geográfica en función de la conexión GPS que traen estos dispositivos, o bien por triangulación entre las antenas de telefonía o WiFi.



IV

Apéndice



A

Presupuesto del Proyecto

Para poder comprender el alcance de este Proyecto Final de Carrera, y tener la suficiente perspectiva, la cual nos va a permitir hacernos eco de las diferentes conclusiones que de él se substraen, se hace necesario la incorporación de este anexo, en el cual se dan datos justificativos de los costes asociados al mismo.

APÉNDICE A. PRESUPUESTO DEL PROYECTO

A.1 Introducción

En el capítulo actual mostraremos cuál ha sido el coste que trae consigo el proyecto que estamos tratando. En primer lugar mostraremos una planificación de las diferentes fases que hemos seguido.

A continuación informaremos sobre los recursos utilizados para el desarrollo de la aplicación, diferenciando entre software, hardware, servicios y personal. Para terminar mostrando las cifras.

A.2 Planificación

Es importante realizar una planificación de los tiempos y plazos estimados para cada una de las diferentes tareas que compone un proyecto. A tal efecto hemos elaborado un Gráfico Gantt para mostrar los tiempos consumidos en la consecución del presente proyecto, elaborado a partir de la herramienta gratuita *Gantt Project*.

Para la elaboración del proyecto se han dividido las tareas en las siguientes fases:

1. Documentación

Aquí se definen las tareas de estudio previo a la puesta en marcha de la creación del proyecto, donde se definen los objetivos del desarrollo, así como la situación actual de la tecnología.

- Definición de objetivos
- Estudio del estado del arte

2. Evaluación de las necesidades

Una vez definido el punto de partida, comenzamos a realizar un estudio de las necesidades a las que debe dar respuesta la aplicación a desarrollar, y para ello estudiaremos los Casos de Uso y los diferentes Requisitos, de Usuario y de Software.

APÉNDICE A. PRESUPUESTO DEL PROYECTO

- Casos de Uso
- Requisitos de Usuario
- Requisitos de Software

3. Diseño del sistema

Con toda la información acerca de lo que debe hacer nuestra aplicación, diseñamos como va a ser programada la aplicación.

- Parte Web
- Parte Móvil

4. Desarrollo del sistema

Una vez diseñado el sistema, se procederá al desarrollo del mismo.

- Parte Web
- Parte Móvil

5. Implantación

Para poder realizar las pruebas de verificación es necesario tener los equipos instalados en los equipos finales, cosa de la que se encarga esta fase.

- Parte Web
- Parte Móvil

6. Pruebas realizadas

Una vez instalada la aplicación, verificamos el sistema para ver que cumple con los objetivos y requisitos marcados al comienzo del proyecto.

- Parte Web
- Parte Móvil

7. Redacción de la memoria En último lugar se elaborará esta memoria, la cual dará cabida a todos los datos necesarios para la comprensión del proyecto.

APÉNDICE A. PRESUPUESTO DEL PROYECTO

A.3 Recursos Utilizados

Para poder llevar a cabo este proyecto se ha hecho uso de diferentes herramientas y servicios, así como se ha necesitado el trabajo humano de diferentes personas, que mostraremos en la Tabla A.3.1.

A.3.1 Herramientas Software

Herramienta	Descripción	Coste
<i>SDK Android</i>	Entorno de programación Android	Gratuito
<i>IDE Eclipse</i>	Entorno de programación	Gratuito
<i>GIMP</i>	Edición gráfica de imágenes	Gratuito
<i>FileZilla</i>	Gestor de conexiones FTP	Gratuito
<i>XAMP</i>	Entorno de trabajo Apache+MySQL+PHP	Gratuito
<i>Navegadores</i>	Navegadores web	Gratuito
<i>WinRAR</i>	Compresor de ficheros	Gratuito
<i>TeXnicCenter</i>	Editor gráfico para \LaTeX	Gratuito
<i>MikTex</i>	Gestor de paquetes \LaTeX	Gratuito
<i>Adobe Reader</i>	Lector de documentos PDF	Gratuito
<i>GhostScript</i>	Lector de documentos <i>PostScript</i>	Gratuito

Tabla A.1: Tabla de Recursos de Software

A.3.2 Herramientas Hardware

En la Tabla A.3.2 podemos apreciar los diferentes equipos físicos que hemos utilizado.

Así mismo, estos dispositivos contarán con un tiempo de vida de cinco años, dato que se reflejará en el coste de estos equipos para el presente proyecto, conociendo que el tiempo de uso de los mismos ha sido cinco meses.

APÉNDICE A. PRESUPUESTO DEL PROYECTO

Herramienta	Descripción	Precio €	Coste €
<i>Ordenador</i>	Ordenador personal de sobremesa	1200,00	100,00
<i>Huawei Ivy</i>	<i>Smartphone</i> con el SO Android	210,00	17,50
<i>Router</i>	<i>Router</i> inalámbrico con conexión ADSL	75,00	6,25

Tabla A.2: Tabla de Recursos de Hardware

■ ■ A.3.3 Servicios

Mostramos los servicios en la Tabla A.3.3 que hemos necesitado para realizar el actual proyecto.

Además, destacar que el uso que se da a estos servicios para el presente proyecto aprovecha el 50 % de los mismos, durante un periodo de algo menos de seis meses, con lo que se tendrá en cuenta este aspecto a la hora de calcular los costes, así como si los pagos de los servicios son mensuales (ADSL) o anuales (alojamiento y registro de dominio).

Servicio	Descripción	Precio €	Coste €
<i>Dominio</i>	Registro de un dominio público	10,00	2,08
<i>Hosting</i>	Almacenamiento web de 400 Mb	35,00	7,29
<i>ADSL</i>	Conexión a internet de alta velocidad	50,00	125,00

Tabla A.3: Tabla de Servicios

■ ■ A.3.4 Personal

Además, mostraremos también el personal que ha intervenido en el proyecto en la Tabla A.3.4.

APÉNDICE A. PRESUPUESTO DEL PROYECTO

Nombre	Categ.	Dedicación	Coste €	Total €
<i>Iria Manuela Estévez Ayres</i>	Ingeniero Senior	1	4300,00	4300,00
<i>Antonio Sánchez de la Fuente Rodríguez</i>	Ingeniero	5	2800,00	14000,00

Tabla A.4: Tabla de Recursos de Hardware

A.4 Presupuesto

Una vez hemos obtenido los costes de las diferentes partes los sumaremos para hallar el total del proyecto, añadir a lo anterior, que hemos calculado unos costes indirectos por valor del 20 % del total del proyecto.

Concepto	Precio Total €
<i>Herramientas Software</i>	0,00
<i>Herramientas Hardware</i>	123,75
<i>Servicios</i>	134,37
<i>Personal</i>	18300,00
<i>Subtotal</i>	18558,12
<i>Costes Indirectos</i>	3711,62
TOTAL	22269,64

Tabla A.5: Tabla de Presupuesto Final

APÉNDICE A. PRESUPUESTO DEL PROYECTO

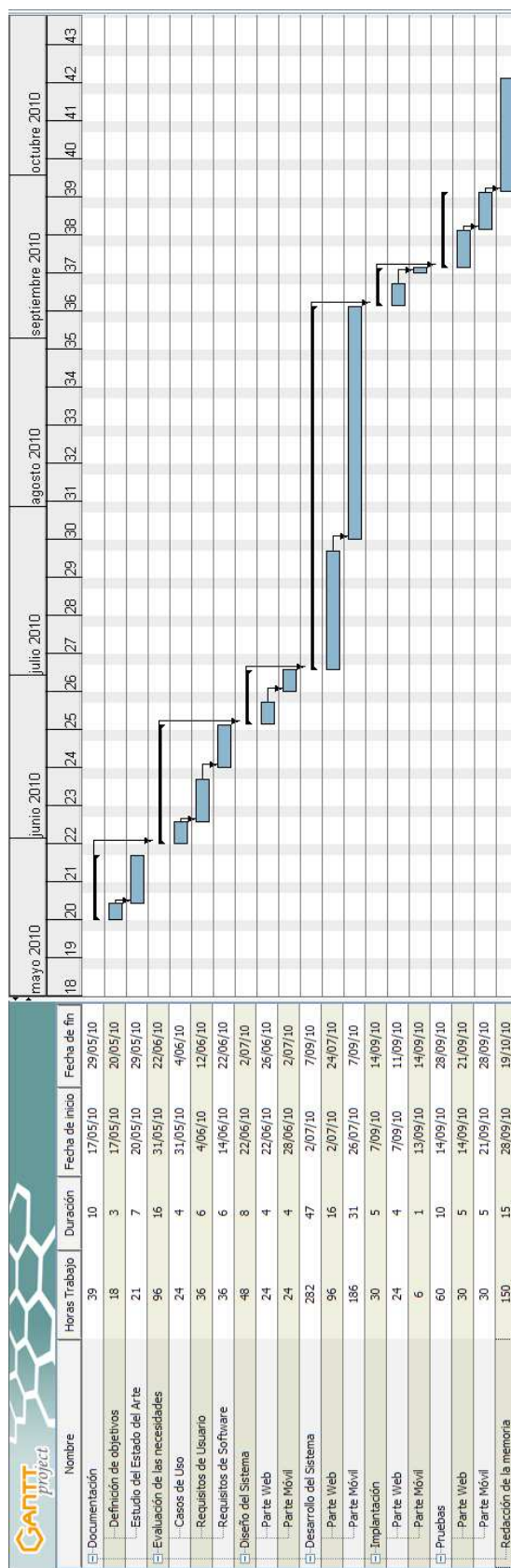


Figura A.1: Gráfico Gantt para la planificación del proyecto



B

Requisitos de Usuario

Para lograr producir aquello que el cliente requiere, en el plazo solicitado y ajustados al presupuesto asignado, se necesita desarrollar un proceso que incluya desde la etapa más temprana la gestión de los requisitos acordados, de forma que se garantice la satisfacción del cliente.

APÉNDICE B. REQUISITOS DE USUARIO

B.1 Introducción

Este capítulo complementa al de Requisitos de la Aplicación, mostrando las diferentes tablas que definen cada Requisito de Usuario, para las dos clases en que se dividen, capacidad y restricción, tanto para la parte web de la aplicación como la parte móvil.

B.2 Tablas de Requisitos de Usuario

■ ■ B.2.1 Requisitos de Usuario de Capacidad

Definiremos así los requisitos del sistema traducidos en funciones y operaciones que dotan a los usuarios de herramientas para lograr obtener el rendimiento deseado de la aplicación, de esta forma describe operaciones que el software ha de llevar a cabo.

- Parte Web

Identificador	RUCW001		
Descripción	Accesibilidad a la plataforma web.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla B.1: Requisitos de Usuario de Capacidad RUCW001

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCW002
Descripción	Login/Logout de usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.2: Requisitos de Usuario de Capacidad RUCW002

Identificador	RUCW003
Descripción	Autologin de usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.3: Requisitos de Usuario de Capacidad RUCW003

Identificador	RUCW004
Descripción	Descargar información.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.4: Requisitos de Usuario de Capacidad RUCW004

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCW005
Descripción	Editar la información del usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.5: Requisitos de Usuario de Capacidad RUCW005

Identificador	RUCW006
Descripción	Ver comentarios introducidos por el usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.6: Requisitos de Usuario de Capacidad RUCW006

Identificador	RUCW007
Descripción	Ver la información asociada a un contenido.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.7: Requisitos de Usuario de Capacidad RUCW007

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCW008
Descripción	Ver los comentarios asociados a un contenido.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.8: Requisitos de Usuario de Capacidad RUCW008

Identificador	RUCW009
Descripción	Editar un comentario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.9: Requisitos de Usuario de Capacidad RUCW009

APÉNDICE B. REQUISITOS DE USUARIO

■ Parte Móvil

Identificador	RUCM001
Descripción	Instalación de la aplicación.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.10: Requisitos de Usuario de Capacidad RUCM001

Identificador	RUCM002
Descripción	Iniciar la aplicación.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.11: Requisitos de Usuario de Capacidad RUCM002

Identificador	RUCM003
Descripción	Salir de la aplicación.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.12: Requisitos de Usuario de Capacidad RUCM003

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCM004
Descripción	Ver Ayuda.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.13: Requisitos de Usuario de Capacidad RUCM004

Identificador	RUCM005
Descripción	Login/Logout de usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.14: Requisitos de Usuario de Capacidad RUCM005

Identificador	RUCM006
Descripción	Autologin de usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.15: Requisitos de Usuario de Capacidad RUCM006

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCM007		
Descripción	Ver contenido.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla B.16: Requisitos de Usuario de Capacidad RUCM007

Identificador	RUCM008		
Descripción	Puntuar contenido.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla B.17: Requisitos de Usuario de Capacidad RUCM008

Identificador	RUCM009		
Descripción	Ver comentario.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla B.18: Requisitos de Usuario de Capacidad RUCM009

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RUCM010
Descripción	Modificar comentario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.19: Requisitos de Usuario de Capacidad RUCM010

B.2.2 Requisitos de Usuario de Restricción

Los requisitos de Usuario de Restricción van a definir restricciones en las que el software es construido y funciona, pero sin alterar los Requisitos de Usuario de Capacidad.

■ Parte Web

Identificador	RURW001
Descripción	Tiempo de respuesta en la carga de nuevas páginas.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.20: Requisitos de Usuario de Restricción RURW001

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RURW002
Descripción	Tiempo de respuesta en las consultas a la base de datos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.21: Requisitos de Usuario de Restricción RURW002

Identificador	RURW003
Descripción	Resolución de pantalla.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.22: Requisitos de Usuario de Restricción RURW003

Identificador	RURW004
Descripción	Independencia de SO.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.23: Requisitos de Usuario de Restricción RURW004

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RURW005
Descripción	Compatibilidad con navegadores.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.24: Requisitos de Usuario de Restricción RURW005

Identificador	RURW006
Descripción	Acceso desde lugares remotos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.25: Requisitos de Usuario de Restricción RURW006

Identificador	RURW007
Descripción	Parámetros iniciales.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.26: Requisitos de Usuario de Restricción RURW007

APÉNDICE B. REQUISITOS DE USUARIO

■ Parte Móvil

Identificador	RURM001
Descripción	Ver comentarios introducidos por el usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.27: Requisitos de Usuario de Restricción RURM001

Identificador	RURM002
Descripción	Respuesta en la carga de la aplicación.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.28: Requisitos de Usuario de Restricción RURM002

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RURM003
Descripción	Respuesta en la carga de nuevos <i>Activity</i> .
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.29: Requisitos de Usuario de Restricción RURM003

Identificador	RURM004
Descripción	Respuesta en la captura de datos web.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.30: Requisitos de Usuario de Restricción RURM004

Identificador	RURM005
Descripción	Resolución de pantalla.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.31: Requisitos de Usuario de Restricción RURM005

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RURM006
Descripción	Uso de SO.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.32: Requisitos de Usuario de Restricción RURM006

Identificador	RURM007
Descripción	Compatibilidad con versiones antiguas.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.33: Requisitos de Usuario de Restricción RURM007

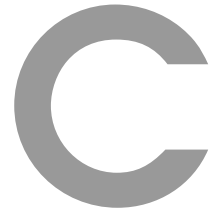
Identificador	RURM008
Descripción	Parámetros configurables.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.34: Requisitos de Usuario de Restricción RURM008

APÉNDICE B. REQUISITOS DE USUARIO

Identificador	RURM009
Descripción	Guardado de información.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla B.35: Requisitos de Usuario de Restricción RURM009



Requisitos de Software

La captura de requisitos es de vital importancia debido a que permite gestionar las necesidades del proyecto en forma estructurada, mejora la capacidad de predecir cronogramas, así como mejora la calidad del software al cumplir los requerimientos acordados, y evita un posible rechazo de los usuarios finales al ajustarse a sus necesidades.

APÉNDICE C. REQUISITOS DE SOFTWARE

C.1 Introducción

Este capítulo complementa al de Requisitos de la Aplicación, mostrando las diferentes tablas que definen cada Requisito de Software, para las clases en que se dividen, Funcionales, Rendimiento, Interfaz, Recursos y Seguridad; tanto para la parte web de la aplicación como la parte móvil.

C.2 Tablas de Requisitos de Software

■ ■ C.2.1 Requisitos de Software Funcionales

Especifican cuál es la funcionalidad que tiene el sistema, de ese modo definen los propósitos del software en cuestión, siendo estos requerimientos derivados de los Requisitos de Usuario especificados en la sección anterior.

■ Parte Web

Identificador	RSFW001		
Descripción	Consulta de información de contenidos.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.1: Requisitos de Software Funcionales RSFW001

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSFW002
Descripción	Reproducción de contenidos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.2: Requisitos de Software Funcionales RSFW002

Identificador	RSFW003
Descripción	Inicio del portal web
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.3: Requisitos de Software Funcionales RSFW003

Identificador	RSFW004
Descripción	Gestión de credenciales.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.4: Requisitos de Software Funcionales RSFW004

APÉNDICE C. REQUISITOS DE SOFTWARE

■ Parte Móvil

Identificador	RSFM001		
Descripción	Consulta de información de contenidos.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.5: Requisitos de Software Funcionales RSFM001

Identificador	RSFM002		
Descripción	Reproducción de contenidos.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.6: Requisitos de Software Funcionales RSFM002

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSFM003
Descripción	Inicio de la aplicación.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.7: Requisitos de Software Funcionales RSFM003

Identificador	RSFM004
Descripción	Inicio de la reproducción.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.8: Requisitos de Software Funcionales RSFM004

Identificador	RSFM005
Descripción	Guardado de credenciales.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.9: Requisitos de Software Funcionales RSFM005

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSFM006
Descripción	Lectura de credenciales.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.10: Requisitos de Software Funcionales RSFM006

■ ■ C.2.2 Requisitos de Software de Rendimiento

Describen límites o condiciones sobre cómo diseñar la aplicación, aunque no deben llegar a suplantar dicho proceso, especificando condiciones impuestas por el cliente, el entorno u otros factores sobre el proyecto.

■ Parte Web

Identificador	RSRW001
Descripción	Tiempo de respuesta a la carga de <i>Activity</i> .
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.11: Requisitos de Software de Rendimiento RSRW001

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSRW002
Descripción	Tiempo de respuesta a la carga de contenidos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.12: Requisitos de Software de Rendimiento RSRW002

Identificador	RSRW003
Descripción	Tiempo de respuesta a consultas a la base de datos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.13: Requisitos de Software de Rendimiento RSRW003

Identificador	RSRW004
Descripción	Conexión a internet.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.14: Requisitos de Software de Rendimiento RSRW004

APÉNDICE C. REQUISITOS DE SOFTWARE

■ Parte Móvil

Identificador	RSWM001
Descripción	Tiempo de respuesta a la carga de <i>Activity</i> .
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.15: Requisitos de Software de Rendimiento RSWM001

Identificador	RSWM002
Descripción	Tiempo de respuesta a la carga de contenidos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.16: Requisitos de Software de Rendimiento RSWM002

Identificador	RSWM003
Descripción	Tiempo de respuesta a consultas a la base de datos.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.17: Requisitos de Software de Rendimiento RSWM003

Identificador	RSRM004
Descripción	Conexión a internet.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.18: Requisitos de Software de Rendimiento RSRM004

■ ■ C.2.3 Requisitos de Software de Interfaz

Desarrolla la forma en que la aplicación se comunica con su entorno, bien en la comunicación con el usuario, o bien en la comunicación con otras aplicaciones.

- Parte Web

Identificador	RSIW001
Descripción	Navegadores.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.19: Requisitos de Software de Interfaz RSIW001

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSIW002		
Descripción	Gestión de la información.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.20: Requisitos de Software de Interfaz RSIW002

■ Parte Móvil

Identificador	RSIM001		
Descripción	Navegadores.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.21: Requisitos de Software de Interfaz RSIM001

Identificador	RSIM002		
Descripción	Gestión de la información.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.22: Requisitos de Software de Interfaz RSIM002

C.2.4 Requisitos de Software de Recursos

Especifican los tipos y la cantidad de recursos que el sistema va a necesitar, desde el uso de memoria hasta el tráfico de datos que soportará.

- Parte Web

Identificador	RSCW001		
Descripción	Memoria utilizada.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.23: Requisitos de Software de Recursos RSCW001

Identificador	RSCW002		
Descripción	Hardware necesario.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.24: Requisitos de Software de Recursos RSCW002

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSCW003		
Descripción	Conexiones necesarias.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.25: Requisitos de Software de Recursos RSCW003

■ Parte Móvil

Identificador	RSCM001		
Descripción	Memoria utilizada.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.26: Requisitos de Software de Recursos RSCM001

Identificador	RSCM002		
Descripción	Hardware necesario.		
Fuente	Desarrollador	Estudio de la Competencia	
Necesidad	Esencial	No Esencial	
Prioridad	Alta	Medio	Baja
Estabilidad	Estable	No Estable	
Verificabilidad	Si	No	

Tabla C.27: Requisitos de Software de Recursos RSCM002

Identificador	RSCM003
Descripción	Conexiones necesarias.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.28: Requisitos de Software de Recursos RSCM003

■ ■ C.2.5 Requisitos de Software de Seguridad

Definen la seguridad del sistema frente a amenazas externas o sobre la propia seguridad de las personas, ejemplos de ello son la confidencialidad, integridad o disponibilidad.

■ Parte Web

Identificador	RSSW001
Descripción	Comprobación de los datos del usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.29: Requisitos de Software de Seguridad RSSW001

APÉNDICE C. REQUISITOS DE SOFTWARE

Identificador	RSSW002
Descripción	Comprobación en el acceso del usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.30: Requisitos de Software de Seguridad RSSW002

■ Parte Móvil

Identificador	RSSM001
Descripción	Comprobación de los datos del usuario
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.31: Requisitos de Software de seguridad RSSM001

Identificador	RSSM002
Descripción	Comprobación en el acceso del usuario.
Fuente	Desarrollador Estudio de la Competencia
Necesidad	Esencial No Esencial
Prioridad	Alta Medio Baja
Estabilidad	Estable No Estable
Verificabilidad	Si No

Tabla C.32: Requisitos de Software de Seguridad RSSM002



D

Bibliografía

Es de vital importancia para el autor de una obra referenciar aquellos escritos que le han ayudado a redactar su documento, cobrando tal importancia este hecho que se ha estandarizado por medio de una norma ISO, ISO 690-1987. Por ello, y porque este Proyecto Final de Carrera no podría haber sido posible sin contar con los documentos que se citan a continuación, hemos elaborado esta bibliografía.

Artículos

- [35] Ramón Muñoz. Windows phone 7 tiene escritorio vivo. *Diario El País*, 2010.
<http://www.elpais.com/>.
- [42] Diario El País Barcelona. Android supera en ventas a iphone. *El País*, 2010.
<http://www.elpais.es/>.
- [44] Agencia Reuters. Samsung ofrecerá el 33 % de sus equipos con bada. *Reuters US*, 2010.
<http://www.reuters.com/article/idUSSEU00305420100430>.

Libros

- [9] E. Narmore M.K. Glass, Y.L. Scouarnec. *Desarrollo Web con PHP, Apache y MySQL*, volume 1. Anaya Multimedia, 2004.
- [10] Ministerio de Asuntos Exteriores y Cooperación. *Guía para la integración de las TIC en la Cooperación Española*, volume 1. Baetica.net, 2006.
- [11] E.K. Lua J.F. Buford, H. Yu. *P2P Networking and Applications*, volume 1. Morgan Kaufman, 2009.
- [12] George Coulouris. *Distributed Systems. Concepts and Design*, volume Third Edition. Addison Wesley, 2001.
- [14] Javier Celaya. *La empresa en la web 2.0*, volume Primera Edición. Gestión 2000, 2008.
- [29] David Austerberry. *The Technology of Audio and Video Streaming*, volume 2. Focal Press - Elsevier, 8-10, rue Mario -Nikis, 75738 PARIS CEDEX, Francia, 2005.
- [30] Andrew S. Tanenbaum. *Redes de Computadores*, volume Cuarta edición. Pearson. Prentice Hall, 2003.
- [32] Comisión del Mercado de las Telecomunicaciones. *Informe Anual 2009*, volume Primera Edición. Comisión del Mercado de las Telecomunicaciones, 2009.
- [55] Agencia Espacial Europea. *Guía para la aplicación de Estándares de Ingeniería de Software ESA (Agencia Espacial Europea) para proyectos de software pequeños. Revisión 2003*. ESA Comité de Estandarización y Control de Software (BSSC), 8-10, rue Mario -Nikis, 75738 PARIS CEDEX, Francia, 1991.
- [58] R. Sen F. Ableson, C. Collins. *Android. Guía para desarrolladores*, volume Primera Edición. Anaya Multimedia, 2009.

Tesis Doctorales

- [7] Sofía García Marina. Máster de estudios avanzados en museos y patrimonio histórico-artístico. Master's thesis, Universidad Complutense de Madrid, 2010.

Recursos electrónicos

- [1] Radar Networks and Nova Spivack. Semantics of social connections. Website, 2010.
<http://www.semanticfocus.com/>.
- [2] ArbigenSYS. Language usage indicators. Website, 2010. <http://lui.arbingersys.com/>.
- [3] AdMob. Estadísticas de smartphones. Website, 2010. <http://www.admob.com/>.
- [4] Android. Portal android native and software development kit. Website, 2010.
<http://developer.android.com/>.
- [5] GTP. Gtp museum solutions. Website, 2010. <http://www.gtp.es/>.
- [6] Diario Android. Comparativa entre sistemas operativos móviles. Website, 2009.
<http://diarioandroid.com/2009/05/06/comparativa-entre-sistemas-operativos-moviles/>.
- [8] Computer History. Historia de las computadoras. Website, 2010.
<http://www.computerhistory.org/>.
- [13] World Wide Web Consortium. Portal español del consistorio w3c. Website, 2010.
<http://www.w3c.es/>.
- [15] World Wide Web Consortium. Html 4.01 specification. Website, 2010.
<http://www.w3.org/TR/REC-html40/>.
- [17] Oracle Sun Developer Networks. Applet documentation. Website, 2010.
<http://java.sun.com/applets/>.
- [18] Adobe. Flash professional home. Website, 2010.
<http://www.adobe.com/products/flash/>.
- [19] World Wide Web Consortium. Css 2 revision 1 specification. Website, 2010.
<http://www.w3.org/TR/CSS2/>.

APÉNDICE D. BIBLIOGRAFÍA

- [20] World Wide Web Consistorium. Cgi specification. Website, 2010.
<http://www.w3.org/CGI/>.
- [21] Perl6 Comunity Development Server. Official perl6 specification. Website, 2010.
<http://perlcabal.org/syn/>.
- [22] Python. Python programming language official website. Website, 2010.
<http://www.python.org/>.
- [23] Microsoft. Standard asp.net web server controls. Website, 2010.
[http://msdn.microsoft.com/en-us/library/tt9b3d0f\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/tt9b3d0f(VS.85).aspx).
- [24] PHP. Php specification. Website, 2010. <http://www.php.net/>.
- [25] Sun Oracle. Javasever pages standard library. Website, 2010.
<http://www.oracle.com/technetwork/java/index-jsp-135995.html>.
- [26] Ruby. Ruby standard library documentation. Website, 2010.
<http://ruby-doc.org/stdlib/>.
- [27] World Wide Web Consistorium. Extensible markup language specification. Website, 2010.
<http://www.w3.org/TR/REC-xml/>.
- [28] World Wide Web Consistorium. Web sql database. Website, 2010.
<http://www.w3.org/TR/webdatabase/>.
- [31] LLC YouTube. Comunidad de vídeo online. Website, 2010. <http://www.youtube.com/>.
- [33] Symbian Foundation. Fundación symbian. Website, 2010. <http://www.symbian.org/>.
- [34] Microsoft. Windows mobile developer center. Website, 2010.
<http://msdn.microsoft.com/en-us/windowsmobile/default.aspx>.
- [36] Blackberry. Portal de blackberry. Website, 2010. <http://es.blackberry.com/>.
- [37] Blackberry. Portal de desarrollo de blackberry. Website, 2010.
<http://es.blackberry.com/developers/>.
- [38] Palm OS. Palm os development. Website, 2010. <http://developer.palm.com/>.

- [39] Intrepidus Group. Fallo de seguridad en webos. video, 2010.
<http://www.youtube.com/watch?v=SfDRSsh-Ccc&translated=1>.
- [40] Apple. ios dev center. Website, 2010. <http://developer.apple.com/devcenter/ios/index.action>.
- [41] Android. Android developers. Website, 2010. <http://developer.android.com/index.html>.
- [43] Samsung. bada development. Website, 2010. <http://developer.bada.com/>.
- [45] MeeGo. Meego developers. Website, 2010. <http://meego.com/developers>.
- [46] LiMo Foundation. Limo foundation developers oportunity. Website, 2010.
<http://limofoundation.org/en/developer-opportunities.html>.
- [47] Oracle. Sun developers networks. Website, 2010.
<http://developers.sun.com/mobility/index.jsp>.
- [48] Toozla. Toozla solutions. Website, 2010. <http://www.toozla.com/>.
- [49] BlueHertz. Call and talk. Website, 2010. <http://www.bluehertz.es/>.
- [50] TourMovil. Guías turísticas tourmovil. Website, 2010. <http://www.tourmovil.com/>.
- [51] Turismo de Argentina. El portal esencial del turismo en argentina. Website, 2010.
<http://www.turismo.com.ar/>.
- [52] Escoitar. Guía binaural notours. Website, 2010. <http://www.escoitar.org/notours/>.
- [53] Babia. Babia audioguías. Website, 2010. <http://www.babiamobile.com/>.
- [54] wikiTude. Imprint mobilize. Website, 2010. <http://www.wikitude.org/>.
- [56] SL Hospedaxes Galicia. Hospedaxes. Website, 2010. <http://www.hospedaxes.com/>.
- [57] The Eclipse Foundation. Eclipse. Website, 2010. <http://www.eclipse.org/>.
- [59] The Apache Friends. Xamp. Website, 2010. <http://www.apachefriends.org/>.
- [60] NetCraft. Estadísticas de servidores web. Website, 2010. <http://news.netcraft.com/>.